© Bruno Müller-Clostermann, Universität Essen
Alle Rechte liegen beim Autor.
Herstellung: Books on Demand GmbH, Norderstedt
ISBN 3-8311-2823-5

Kursbuch Kapazitätsmanagement

Kompendium für Planung, Analyse und Tuning von IT-Systemen

Herausgegeben von

Bruno Müller-Clostermann

Institut für Informatik
Universität Essen

VORWORT UND DANKSAGUNGEN

In den Jahren 1998 bis 2001 wurde das Verbundprojekt MAPKIT vom Bundesministerium für Bildung und Forschung (BMBF) gefördert unter Projektträgerschaft der DLR (Projektträger IT – Informatiksysteme, Berlin-Adlershof). Kooperationspartner waren die Siemens AG (zunächst als Siemens Nixdorf, nach Umorganisationen als Fujitsu Siemens Computers und Siemens Business Services), das Software- und Systemhaus Materna Information & Communications in Dortmund und das Institut für Informatik an der Universität Essen.

Das Projektziel bestand darin, eine problemadäquate und fortschreibbare Methodik für das Kapazitätsmanagement von verteilten Systemen zu entwickeln, zu erproben und dauerhaft zu etablieren. Dies geschah unter Nutzbarmachung sowohl bekannter und teilweise weiter entwickelter Verfahren und Tools. Ablauf und Ergebnisse des Projekts werden durch die hier zusammengestellten Beiträge in Form eines Handbuchs dokumentiert. Die Beiträge sind inhaltlich breit gestreut und sollen in diesem vielseitigen und komplexen, aber extrem wichtigen Gebiet zur zielführenden Orientierung verhelfen, daher der Titel „Kursbuch Kapazitätsmanagement“.

Mein Dank als Herausgeber gilt allen Autorinnnen und Autoren sowie ganz besonders Frau Anne Fischer für das Korrekturlesen und ihren unermüdlichen Kampf mit den Eigenarten des verwendeten Textsystems.

Dank schulden alle Projektpartner dem BMBF für die Projektförderung und dem DLR für die Projekträgerschaft. Namentlich sind hier zu nennen Herr Abendroth vom BMBF, sowie von der DLR die Herren Dr. Schmidt und Herentz für ihren Einsatz in der schwierigen Vorprojekt- und Startphase und last but not least Frau Dr. Grothe, die während der Projektlaufzeit die Nachfolge von Herrn Dr. Schmidt übernahm.

Bruno Müller-Clostermann Essen, im September 2001

Prof. Dr. Bruno Müller-Clostermann
Institut für Informatik, Universität Essen
Schützenbahn 70 (Hochhaus)
45117 Essen

bmc@informatik.uni-essen.de

INHALT

TEIL I: EINFÜHRUNG UND GRUNDLAGEN

01 **Kapazitätsmanagement gestern & heute I-3**
R. Bordewisch, C. Flues, R. Grabau, J. Hintelmann, K. Hirsch, H. Risthaus

02 **Anforderungen an das Kapazitätsmanagement I-16**
R. Bordewisch, C. Flues, R. Grabau, J. Hintelmann, K. Hirsch, H. Risthaus

03 **Methodik und Verfahren I-26**
R. Bordewisch, C. Flues, R. Grabau, J. Hintelmann, K. Hirsch, F.-J. Stewing

04 **Werkzeuge I-62**
R. Bordewisch, C. Flues, R. Grabau, J. Hintelmann, K. Hirsch, F.-J. Stewing

05 **Praktische Umsetzung I-93**
R. Bordewisch, C. Flues, R. Grabau, J. Hintelmann, K. Hirsch, F.-J. Stewing

06 **Herausforderungen und Chancen I-103**
Kathrin Sydow

TEIL II: MESSUNG UND MONITORING

01 **Das Mess- und Auswertesystem PERMEV II-4**
Bärbel Schwärmer

02 **Proaktives Kapazitätsmanagement II-10**
Reinhard Bordewisch, Kurt Jürgen Warlies

03 **Application Expert II-17**
Klaus Hirsch

04 **Service Level Management II-21**
Stefan Schmutz, Norbert Scherner

05 **Universeller Lastgenerator SyLaGen II-33**
Reinhard Bordewisch, Bärbel Schwärmer

06 **Anforderungen an Messwerkzeuge II-37**
Reinhard Bordewisch, Jörg Hintelmann

TEIL III: MODELLIERUNG UND PROGNOSE

01 **Prognosemodelle - Eine Einführung III-4**
Bruno Müller-Clostermann

02 **The Tool VITO: A short survey III-10**
Michael Vilents, Corinna Flüs, Bruno Müller-Clostermann

03 **Modellexperimente mit VITO III-17**
Bruno Müller-Clostermann, Michael Vilents

04 **Das Werkzeug EcoPREDICTOR III-31**
Torsten Bernert, Bruno Müller-Clostermann

05 **Das SimulationsTool COMNET-III III-43**
Torsten Bernert, Jörg Hintelmann

TEIL IV: MANAGEMENT FRAMEWORKS

01 **CA Unicenter IV-3**
Heiner Risthaus

02 **HP Tools IV-11**
Heiner Risthaus, Ronald Schaten

03 **GlancePlus IV-20**
Josef Berschneider

04 **BEST/1 - Eine Übersicht IV-24**
Jürgen Lührs

05 **BEST/1 - Ein Erfahrungsbericht IV-32**
Jürgen Lührs

TEIL V: WEB UND INTRANET

01 **Das Werkzeug WebLOAD V-3**
Norbert Scherner

02 **Untersuchung des Web-Accounting V-9**
Norbert Scherner

03 **Client/Server Performance-Analyse V-12**
Corinna Flüs

TEIL VI: SAP R/3

01 **SAP R/3 Kapazitätsmanagement – Grundlagen und Werkzeuge im Überblick VI-3**
Michael Paul, Kay Wilhem

02 **Monitoring und Benchmarking für das R/3-Performance -Engineering VI-16**
Kay Wilhelm, Jürgen Pfister, Christian Kowarschick, Stefan Gradek

03 **Modellierung und Prognose für SAP R/3 mit dem WLPsizer VI-48**
Kay Wilhelm

TEIL VII: PRAXISBERICHTE

01 **Kapazitätsplanung von IT-Systemen VII-4**
Siegfried Globisch, Klaus Hirsch

02 **Das MAPKIT-Vorgehensmodell: Kapazitätsplanung eines heterogenen Client/Server-Systems VII-9**
Corinna Flues, Jörg Hintelmann

03 **e-Managementleitstand "Light" VII-30**
Achim Manz-Bothe, Mathias Hehn, Franz-Josef Stewing

04 **Mit Last- und Performance-Tests zum einsatzreifen System VII-38**
Hadi Stiel

05 **Last- und Performance-Tests am Beispiel IT2000 VII-44**
Reinhard Bordewisch, Michael Lücke

TEIL VIII: QUELLEN

01 **Referenzen VIII-3**

02 **Ressourcen im World Wide Web VIII-6**

03 **Autoren VIII-7**

04 **Kontaktadressen VIII-10**

Teil I: Einführung und Grundlagen

Inhalt von Teil I

KAPITEL 01 KAPAZITÄTSMANAGEMENT GESTERN & HEUTE 3

01.01 MOTIVATION 3
01.02 METHODEN, VERFAHREN UND WERKZEUGE 4
01.03 FALLBEISPIELE 7
01.04 MISSVERSTÄNDNISSE, UNTERLASSUNGSSÜNDEN UND DENKFEHLER 12

KAPITEL 02 ANFORDERUNGEN AN DAS KAPAZITÄTSMANAGEMENT 16

02.01 KAPAZITÄTSMANAGEMENT: ZIELE UND INHALTE 16
02.02 GENERELLE ANFORDERUNGEN AN DAS KAPAZITÄTSMANAGEMENT 18

KAPITEL 03 METHODIK UND VERFAHREN 26

03.01 MAPKIT-VORGEHENSMODELL 26
03.02 VERFAHREN 48

KAPITEL 04 WERKZEUGE 62

04.01 MESS- UND MONITORING-WERKZEUGE 62
04.02 WERKZEUGE ZUR PLANUNG UND PROGNOSE 78
04.03 ZUSAMMENSPIEL VON MESS- UND PROGNOSE-WERKZEUGEN 89

KAPITEL 05 PRAKTISCHE UMSETZUNG 93

05.01 ERFAHRUNGEN BEI DER KAPMAN-PROZESSETABLIERUNG 93
05.02 BEISPIEL PROJEKT ATLAS 93
05.03 KAPMAN-TRAINING UND WEITERQUALIFIKATION 96
05.04 VORBEREITUNG ZUM ERSTGESPRÄCH BEI ANWENDERN 96
05.05 DETAILLIERTE FRAGENSAMMLUNG 98

KAPITEL 06 HERAUSFORDERUNGEN UND CHANCEN 103

06.01 UMGANG MIT KOMPLEXEN SITUATIONEN IM IT-BEREICH 103
06.02 WAS IST KAPAZITÄTSMANAGEMENT? 105
06.03 BRIDGING THE GAP – MEHR ALS WORTE 106
06.04 VOM PRINZIP ZUM HANDELN 106
06.05 KAPAZITÄTSMANAGEMENT ALS RETTER IN DER NOT? 107
06.06 DIE TÄGLICHEN HÜRDEN – ZWISCHEN BEGEISTERUNG UND ABLEHNUNG 108
06.07 KEINE LEICHTE AUFGABE 109
06.08 VIRTUELLES TEAM – NUR EIN FROMMER WUNSCH? 109
06.09 KAPAZITÄTSMANAGEMENT UND DAS FÖRDERPROJEKT MAPKIT – ESOTERISCHE SPIELEREI ODER NOTWENDIGE INNOVATION? 110

KAPITEL 01 KAPAZITÄTSMANAGEMENT GESTERN & HEUTE

R. BORDEWISCH, C. FLUES, R. GRABAU, J. HINTELMANN,
K. HIRSCH, H. RISTHAUS

01.01 Motivation

Trotz der zunehmenden Abhängigkeit des Unternehmenserfolgs von der Leistungsfähigkeit und der Verlässlichkeit der IT-Strukturen ist ein systematisches Kapazitätsmanagement (KapMan) in vielen Unternehmen völlig unzureichend etabliert. Betrachtet man die einschlägige Literatur, so lassen sich die Begriffe Kapazitätsmanagement und Kapazitätsplanung wie folgt abgrenzen:

Unter Kapazitätsmanagement versteht man alle Aktivitäten, die darauf abzielen, die vorhandenen IT-Ressourcen so einzusetzen, dass sich eine bestmögliche Dienstgüte und Ressourcennutzung für alle Anwendungen einstellt. Zu diesen Aktivitäten zählen Anpassung von Anwendungsgewohnheiten, Umkonfigurieren von Systemen und Einstellungen von Systemparametern zur Erreichung maximaler Leistung. Letzteres wird auch als Performance Tuning bezeichnet. Daher kann man Kapazitätsmanagement als Optimierungsprozess gegenwärtiger Systeme auffassen.

Kapazitätsplanung umfasst alle Aktivitäten, um für vorgegebene Dienstgüteanforderungen von Anwendungen die notwendigen Ressourcen bereitzustellen. Dazu zählen die Aufrüstung von Client-Rechnern, Servern und Netzkomponenten, die Anschaffung neuer Komponenten oder auch die Bildung von Substrukturen, um Verkehrsströme voneinander zu trennen. Kapazitätsplanung ist ein auf zukünftige Systeme ausgerichteter Prozess.

Die Betrachtung des Kapazitätsmanagements als reinen Optimierungsprozess erscheint uns zu kurz gefasst. Dabei ist Management nicht nur im Sinne des Führens, sondern mehr im Sinne des Handhabens und des Bewerkstelligens zu verstehen. Daher führen wir die folgende Sichtweise ein: KapMan umfasst die Aktivitäten Überwachen, Analysieren und Tunen, Planen und Prognostizieren des Systemverhaltens hinsichtlich Performance, Zuverlässigkeit und Verfügbarkeit.

IT-Anwender sind an einem performanten, stabilen Systemverhalten und IT-Betreiber an einer wirtschaftlichen Ressourcen-Nutzung interessiert. Deshalb bilden vertragliche Vereinbarungen über die gewünschte Dienstgüte wie Transaktionsdurchsatz oder Antwortzei-

ten in Form von ”Service Level Agreements”, wie auch über die Kosten von nachweislich verbrauchten IT-Ressourcen einen wesentlichen Aspekt des KapMan-Prozesses.

01.02 Methoden, Verfahren und Werkzeuge

Die durchgängige Umsetzung der KapMan-Methodik ist heute noch sehr stark plattformabhängig bzw. anwendungsbezogen:

(a) Plattformen

BS2000

Im homogenen Mainframe-Umfeld ist ein systematisches KapMan etabliert und wird in vielen Fällen sowohl vom eigenen Vertrieb als auch von Anwendern akzeptiert. Die methodische Unterstützung reicht von Performance-Handbüchern als Hilfsmittel für verantwortliche Systemadministratoren bis hin zu einer Reihe von Verfahren und Werkzeugen, die die KapMan-Aktivitäten unterstützen. Dazu zählen insbesondere spezielle Mess- und Auswerteumgebungen sowie plattformspezifische Prognosewerkzeuge.

Überwachung, Analyse und Tuning

Zur dynamischen Regelung der Last kommen steuernde und performancesteigernde Werkzeuge (z.B. PCS, SPEEDCAT, DAB) zum Einsatz.

In der Überwachung und Analyse werden sowohl auf System- als auch auf Anwendungs- und Datenbankebene verschiedene Monitore und Auswerter mit unterschiedlicher Detaillierungstiefe eingesetzt (SM2, COSMOS, KDCMON, SESCOS).

Planung und Prognose

Für Planungen wird das ausgereifte Modellierungswerkzeug COPE eingesetzt. COPE übernimmt die Werte für die Modellinputparameter zur Beschreibung von Last und Hardware automatisch aus COSMOS-Messungen. Der Einsatz spezieller Lasttreiber (SIMUS, FITT) ermöglicht die reproduzierbare Abwicklung kundenspezifischer Benchmarks.

UNIX

In diesem Umfeld findet ein systematisches Kapazitätsmanagement kaum statt, oft begründet durch die (angeblich) gute und hohe Skalierbarkeit der Systeme, gespiegelt an den Ergebnissen der Standard Benchmarks (SPEC, TPC, etc.).

Im Bereich Performance Analyse und Tuning ist eine Vielzahl von Empfehlungen zur Parametrisierung vorhanden, die auf Erfahrungen im täglichen Einsatz basieren (Tuning-

Leitfäden). Messwerkzeuge sind in großer Anzahl vorhanden. Der Umfang sowie die Messmöglichkeiten sind jedoch stark abhängig von der aktuellen UNIX-Plattform.

Überwachung, Analyse und Tuning

Für Einzelkomponenten sind standardmäßig Messwerkzeuge vorhanden (SAR, dkstat, netstat, ...). Um jedoch ein Bild über das Gesamtsystem zu erhalten, müssen diese synchronisiert und die erhobenen Messdaten korreliert werden. Beispiele solcher Umgebungen sind ViewPoint, GlancePlus und PERMEX. Allerdings bleibt anzumerken, dass auch bei Einsatz dieser Umgebungen es auf UNIX-Plattformen nicht möglich ist, mit systemseitig verfügbaren Mitteln im laufenden Betrieb das Antwortzeitverhalten von Benutzerdialogen, die zugehörigen Dialograten und Ressourcen-Verbräuche dieser Dialoge zu erheben.

Planung und Prognose

Planungswerkzeuge für UNIX-Systeme sind ausreichend vorhanden. Beispiele sind BEST/1 und Athene. Ein großes Problem stellt hier die fehlende Unterstützung des Baselining, also der Überführung von Messdaten in Modelleingangsparameter, dar. Als weitere Verfahren zur Planung im UNIX-Umfeld werden Lastsimulationen und Kunden-Benchmarking verwendet.

Windows

Hier wird KapMan so gut wie gar nicht angewendet, denn diese Systeme kommen aus der sog. ”PC-Schiene”, wo die sog. ”Nachschiebementalität” vorherrscht: Bei Bedarf rüstet man einfach auf.

Überwachung, Analyse und Tuning

Für die Überwachung und Analyse des Systemverhaltens bietet Windows mit den Standard-Software-Monitoren perfmon (Systemmonitor) und perflog (Performance Data Log Services) zwei gute Messwerkzeuge an, die sowohl Online-Überwachung als auch aufgrund ihrer Konfigurierbarkeit sehr umfassende Ursachenanalysen ermöglichen. Doch analog zu UNIX fehlen auch hier Bezug zu Antwortzeit und zum Transaktionsaufkommen.

Planung und Prognose

Zur Unterstützung des Sizing von Windows-Systemen können allgemeine Modellierungswerkzeuge eingesetzt werden. BEST/1 steht wie im UNIX-Umfeld auch als Werkzeug zur Performance-Analyse und -Prognose zur Verfügung.

Netze/Netzwerk-Betriebssysteme

Beispielhaft seien hier NFS, AS/X, LAN-Manager, NetWare und Samba genannt. Im Netzumfeld findet ein systematisches Kapazitätsmanagement nicht statt. Hier spielen eher Feh-

lermanagement und Lebendigkeitsanalyse zur funktionalen Korrektheit die entscheidende Rolle. Performance-Engpässe werden durch den Einkauf von Bandbreite „gelöst“. Langfristige Planung findet so gut wie nicht statt.

Überwachung, Analyse und Tuning

Im Bereich der Netzwerküberwachung hat sich mit SNMP (Simple Network Management Protocol) ein Industriestandard durchgesetzt. SNMP bietet standardmäßig nur Low-Level-Informationen an und stellt für das Performance-Management Kommunikationsmechanismen bereit. Die meisten Netzkomponenten verfügen über RMON- (Remote Monitoring-) Probes, die über SNMP angesprochen werden. SNMP bietet den Rahmen für die Einbringung von Erweiterungen, z.B. Programmierung eigener Agenten oder RMON-Filter, um ausgewählte Pakete zu analysieren. Zur Analyse des Netzverkehrs werden Analyzer unterschiedlicher Leistungsstärke eingesetzt.

Planung und Prognose

Zur Unterstützung der Dimensionierung von Netzwerken werden etliche kommerziell erwerbbare Prognosewerkzeuge angeboten, die auf dem Paradigma der Warteschlangennetze basieren. Die bekanntesten Werkzeuge sind COMNET III, Strategizer und die OPNet-Tools, die auch Schnittstellen zur automatischen Datenübernahme aus Frameworks (u.a. HP-OpenView, CA-UniCenter, Tivoli) haben. Häufig wird von Anwendern von Client/Server-Installationen der Nachweis einer performanten Verarbeitung einer bestimmten Last auf der angebotenen Konfiguration verlangt. Um zu realistischen Aussagen bezüglich vorgegebener Anwender-Lastprofile zu gelangen, werden Lastgeneratoren (z.B. FIPS) eingesetzt, die die vorgegebene Last reproduzierbar erzeugen und in einer realen oder Testumgebung auf das Netzwerk und die Server-Systeme bringen.

(b) Anwendungen

Datenbanken

Für Standard-Datenbanksysteme wie z.B. Oracle und Informix werden i.d.R. umfangreiche Überlegungen zum funktionalen Design angestellt und es wird der benötigte Plattenplatz ermittelt. Diese DB-Systeme führen etliche Statistiken über Anzahl und Art sowie Verweilzeiten der DB-Aufträge und der Hitraten für die DB-Puffer. Darüber hinaus bieten sie auch häufig Accounting- und Überwachungsfunktionalitäten an, was einen Bezug zwischen Benutzereingabe, dem damit verbundenen Ressourcenbedarf und dem Antwortzeitverhalten auf Server-Ebene liefert. Bedauernswerterweise beschränken sich diese Informationen auf die DB-Systeme und berücksichtigen nicht die Gesamtsystemsicht. Häufig werden diese Informationen Frameworks (z.B. Tivoli, HP OpenView, CA UniCenter, BMC Patrol) und anderen Mess-/Auswerteumgebungen (vgl. PERMEX/PERMEV) zur Verfügung ge-

stellt. Eine prinzipielle Sensibilität bzgl. des Sizing-Prozesses ist vorhanden, aber eine durchgängige KapMan-Vorgehensweise wird kaum angewendet.

SAP R/3

SAP R/3 ist eine vorbildlich instrumentierte Anwendung, die jede Aktivität der Anwendung in Form eines umfangreichen Statistiksatzes protokolliert.

Diese Daten sind die Grundlage für eine Vielzahl von Monitoring- und Auswertetools, die in die Anwendung integriert sind. Somit ist es für einen R/3-Administrator nicht zwingend notwendig, sich mit spezifischen Betriebssystem- oder Datenbanktools auseinander zu setzen.

Auswertungen über das gesamte System – über Rechnergrenzen hinweg – stehen im Vordergrund. Beispielsweise werden Antwortzeiten bis hin zum Endanwender gemessen und detailliert protokolliert, wie sie sich zusammensetzen (Zeiten auf Netz, Applikationsserver und Datenbankserver).

Schnittstellen ermöglichen den Einsatz externer Tools, wie den R/3 Live Monitor.

Das Thema Sizing von Neusystemen hat im SAP R/3-Umfeld einen sehr hohen Stellenwert, basiert aber in der Regel auf den Erkenntnissen aus den R/3 Standard Application Benchmarks und berücksichtigt Customizing-Anpassungen und Eigenentwicklungen des Kunden nur sehr eingeschränkt.

Kundenspezifische Lasttests sind etabliert und werden häufig durchgeführt.

Abschätzungen zukünftiger Systeme auf Basis von Analysen laufender Systeme gewinnen immer mehr an Bedeutung. Ein umfassendes Kapazitätsmanagement allerdings im Sinne des später beschriebenen Vorgehensmodells, das auch eine Modellierung einschließt, ist nach wie vor der Ausnahmefall.

01.03 Fallbeispiele

Die folgenden Fallbeispiele stellen pragmatische Lösungsansätze dar, die im Rahmen des MAPKIT-Projekts verwendet wurden.

(a) Beispiel 1: R/3-Kapazitätsmanagement

R/3 ist eine Standardsoftware mit einem relativ hohen Normierungsgrad, wodurch prinzipiell ein großes Potenzial für automatisierte Datenerfassung, Datenanalyse, Modellerstellung und Prognoserechnung gegeben ist. Grundlegende Voraussetzung ist natürlich nicht nur die Verfügbarkeit sondern auch der zielgerichtete Einsatz von Monitoren sowohl für die

R/3-Anwendungen als auch für die HW/SW-Konfiguration. Einen Einblick in den Stand der Praxis gibt die folgende Fallstudienskizze (ca. 1999).

Ein mittelständisches Unternehmen im deutschsprachigen Raum, welches sich in seiner speziellen Branche als Weltmarktführer etabliert hat, betreibt ein R/3-System mit mehreren Applikationsservern und einem Datenbankserver. Die Hardware-Installation umfasst mehrere 4-Prozessormaschinen unter Windows NT. Die Performance ist gut, das System läuft "rund", allerdings sei ein gelegentliches "Knistern" der Datenbank zu vernehmen. Die Unternehmensleitung hat Expansionspläne, welche zu erhöhtem Lastaufkommen führen werden. Außerdem soll ein neues R/3-Modul eingeführt werden. Dies führt zu der zentralen Frage, ob künftig Windows NT noch eine ausreichende performante Plattform darstellt.

Diese Fragestellung wurde - vorbildlich frühzeitig - seitens des Vertriebs an ein KapMan-Team weitergeleitet. Nach Installation und Parametrisierung des R/3 Live Monitors (heutiger Name: myAMC.LNI) via Telefonsupport (dies war wichtig wegen der großen räumlichen Distanz) und gleichzeitiger Aktivierung des Windows-NT-Systemmonitors waren Messdaten verfügbar. Die Analyse der IST-Situation umfasste u.a. folgende Punkte.

- Aggregierung der sehr umfangreichen R/3-Analysedaten von Dialogschrittebene auf ein erweitertes Workloadprofil, welches im Stundenmittel für jeden Tasktyp (Dialog, Update, Batch, Others) und jede Komplexitätsklasse (sehr gering, ..., sehr komplex) die Ressourcenverbräuche (CPU-Time, DB-Ressourcen, bewegte KByte) und die Antwortzeiten darstellt. Dadurch wird ein guter Überblick über die Situation in dem gewählten Hochlastintervall gegeben.
- Ein zentraler Performance-Indikator für den Dialogbetrieb ist der Anteil der Datenbankzeit an der Antwortzeit. Da bei einigen Tasktypen der DB-Anteil >50% war, wurde eine Filterung nach den Hauptressourcenverbrauchern durchgeführt. Insbesondere die sog. Z-Transaktionen (im Rahmen des Customizing hinzuprogrammierte ABAP-Anwendungen) wurden als Kandidaten für ein Performance-Tuning ermittelt.
- Für ein festgelegtes Zeitintervall (eine Stunde, z.B. 10:00-11:00 Uhr) wurde die Auslastung sowohl der Server und als auch der I/O-Systeme betrachtet. Erwartungsgemäß waren hier keine Probleme erkennbar, auch im Hochlastintervall waren noch genügend Leistungsreserven vorhanden. Einzige Besonderheit war die wegen fehlerhafter Betriebssystemkonfigurierung mangelhafte Hauptspeichernutzung des DB-Servers. Mehrere GByte RAM, welche physikalisch installiert waren, konnten erst nach NT-Neukonfigurierung genutzt werden.

Der erstellte Report wurde dem Vertrieb ca. 2 Wochen nach Durchführung der Messungen zur Verfügung gestellt und hat offenbar geholfen, die weitere Entwicklung zu planen bzw. punktuelle Verbesserungen durchzuführen, wie z.B. das o.g. RAM-Problem am DB-Server zu lösen. Eine Prognosemodellierung war geplant, kam aber nicht zustande. Obwohl der

Kunde offensichtlich langfristig plant und die Bedeutung des Themas Kapazitätsmanagement kennt, wurde der R/3 Live Monitor wieder deaktiviert, weil kurzfristig Ressourcen für einen Printserver benötigt wurden. Einer kontinuierlichen Performanceüberwachung als Grundlage eines systematischen Kapazitätsmanagements wurde somit die Grundlage entzogen.

Als Fazit kann festgehalten werden: Es sind zwar als kurzfristige Maßnahme die Dienste eines KapMan-Teams über 1-2 Monate hinweg genutzt worden, eine langfristige Etablierung des Themas Kapazitätsmanagement ist aber nicht gelungen. Insbesondere gab es kein direktes Feedback vom Kunden zum KapMan-Team.

(b) Beispiel 2: Proaktives Kapazitätsmanagement

Hochverfügbarkeit, Zuverlässigkeit und Dienstgüte sind wichtige Kriterien einer modernen IT-Landschaft. Wie werden diese Kriterien messbar und damit überprüfbar und abrechenbar? Wie können diese Anforderungen sichergestellt werden?

Ein bereits früher verfolgter Ansatz ist die Überwachung der Server-Ressourcen mittels PERMEV. Ein inzwischen immer mehr ins Blickfeld gerückter Aspekt ist das Systemverhalten am Client, wie es sich dem Benutzer gegenüber präsentiert und damit auch die Verweilzeit im Netzwerk beinhaltet.

Subjektiven Äußerungen der Anwender über ihr langsames System können IT-Betreiber nur selten etwas entgegensetzen. Heutige Netzwerkmanagement-Tools basieren auf der Ermittlung der Auslastungsgrade der System- und Netzwerk-Komponenten. Überwachungsmechanismen innerhalb der Anwendungen sind selten oder fehlen ganz. Konkret bemängelten die Endanwender[1] die unbefriedigende Dienstgüte (das schlechte Antwortzeitverhalten) der Bearbeitung ihrer Geschäftsprozesse, so dass der Leiter des Rechenzentrums den objektiven Nachweis der Dienstgüte verlangte.

Dies war die Motivation für die Entwicklung des Referenz-PCs unter MS-Windows. In der realen Infrastruktur werden auf einem exklusiv verfügbaren Client-PC business-kritische Applikationen zyklisch zum Ablauf gebracht, wobei der Endbenutzer mittels eines automatischen Testtreibers simuliert. Die realen Eingaben (Tastatur-Eingaben, Mausklicks) werden mit Hilfe von Visual Test-Skripten nachgebildet, und so auch die relevanten Antwortzeiten ohne Modifikation der zu überwachenden Applikation und ohne Rückkoppelungseffekte auf das Gesamtsystem ermittelt. Mit dem Einsatz des Referenz-PC werden die folgenden Zielsetzungen verfolgt:

---

[1] Im konkreten Fall bei einer Bundesbehörde.

- @ Das Antwortzeitverhalten beliebiger Applikationen in Client/Server-Umgebungen aus Benutzersicht wird messbar und objektiv qualifizierbar.
- @ Nach Kalibrierung der Schwellwerte erkennt der Referenz-PC einen (drohenden) Leistungsengpass früher als der Anwender und stellt dem Administrator diese Information zur Verfügung.
- @ Trendanalysen über die vermessenen Transaktionen ermöglichen eine planerische Reaktion auf Leistungseinschränkungen.

Das Proaktive Kapazitätsmanagement auf Basis der Koppelung des Referenz-PC mit der bewährten PERMEV-Langzeitüberwachung (s.u.) bietet zusätzliche Vorteile.

- @ Die PERMEV-Langzeitüberwachung ermittelt die Ressourcenauslastung am Server und stellt eine Trendanalyse über einen vorzugebenden Zeitraum bereit. Die Messintervalle sind dabei relativ groß gewählt, um das System nicht unnötig zu belasten.
- @ Der Referenz-PC beinhaltet einen SW-Treiber („Testautomat"), der einen realen Benutzer an der grafischen Oberfläche simuliert. Business-kritische Transaktionen werden in festgelegten Abständen durchgeführt und deren Antwortzeit DIN 66273-konform ermittelt und protokolliert. Für jede Transaktion kann ein Schwellwert (= maximal tolerierbare Dauer der Transaktion; siehe Service Level Agreement (SLA) zwischen IT-Betreiber und Fachabteilung) definiert werden.
- @ Bei Schwellwertüberschreitungen am Client wird neben einem Alert an den Administrator auch zusätzlich das PERMEV-Messintervall verkürzt. Man erhält so eine höhere Granularität der Messdaten. Durch die Korrelation der Messdaten mit den Antwortzeiten wird eine gezielte Ursachenanalyse des Server-Systems ermöglicht. Gleichzeitig werden für einen kurzen Zeitraum die Client-Transaktionen in verkürztem Abstand durchgeführt, um das Antwortzeitverhalten bis zu einer „Normalisierung" detailliert zu verfolgen.

Der Nutzen für den IT- und Applikationsbetreiber liegt in der Option, frühzeitig Maßnahmen einleiten zu können, bevor sich ein Anwender beschwert. Mögliche schnelle Aktionen sind die Zuschaltung von Leistungsressourcen (Server, Netz, Applikation, verteilte Ressourcen, ...) oder Abschalten nicht zeitkritischer Hintergrundlasten.

Somit ermöglicht das Proaktive Kapazitätsmanagement ein Agieren statt Reagieren: Von der Problemerkennung zur Problemerkennung und -vermeidung!

(c) Beispiel 3: e-Management-Leitstand

Durch die derzeit stattfindende Rezentralisierung und Konsolidierung entsteht in den Rechenzentren, Serverfarmen und Application Service Providern ein Management- und Überwachungsaufwand, der unter Einsatz von klassischen Tools hohen Personal- und Qualifizierungsbedarf nach sich zieht.

Proprietäre, meist nicht in vorhandene Infrastruktur zu integrierende Einzellösungen verursachen hohe Hardware-, Software- und Schulungskosten. Durch plattformspezifische Lösungen koexistieren häufig unterschiedliche Managementplattformen nebeneinander.

Die Leitstände der Rechenzentren und Provider werden immer komplizierter. Falsch interpretierte oder aufgrund der Anzeigenflut nicht beachtete Anzeigen bedeuten Kosten, Ausfallzeiten und den Verstoß gegen Service Level Agreements. Die für den unmittelbaren Rechnerbetrieb Verantwortlichen müssen eine immer höhere Qualifikation aufweisen.

Ein an die Leitstände kritischer Systeme der klassischen Industrie angelehntes Managementsystem muss sich an den Bedürfnissen des überwachenden Personals ausrichten. Insbesondere muss dem Betreiber ein klarer, Fehlinterpretationen vorbeugender Eindruck vom Zustand der vielfältigen Betriebsparameter vermittelt werden. Jedem Mitarbeiter soll genau die Menge von Informationen dargestellt werden, die in seinen Verantwortungsbereich fällt. Eine aufwändige Schulung soll für den täglichen Betrieb aber auch für die Konfiguration des Systems nicht erforderlich sein.

Die identifizierten Anforderungen Plattformunabhängigkeit und einfache Installation legen die Verwendung einer webzentrierten Architektur nahe. Clientseitig stellen HTML-Seiten mit eingebetteten Java-Applets die zu visualisierenden Informationen dar.

Durch die Entscheidung zum Einsatz von HTML und Java ist automatisch jeder Standardarbeitsplatz als Einsatzort des Leitstandes vorbereitet.

Die auf der Clientseite eingeführte Flexibilität wird auf der Serverseite durch den Einsatz von Java weitergeführt. Für jeden überwachten Systemparameter wird ein Modul eingesetzt; diese Module sind unabhängig voneinander und können getrennt voneinander eingesetzt, erweitert und konfiguriert werden. Jedes Modul läuft auf dem oder den Managementserver(n) als eigenständiger Prozess. Das vorgestellte Architekturmodell wurde als Prototyp zur Überwachung und Visualisierung folgender Parameter realisiert.

- Erreichbarkeit im Netz (generisch für alle Plattformen per ICMP)
- Auslastung von CPU und Hauptspeicher in %
- Umgebungstemperatur, Hardwarestörungen (Stromversorgung, Temperatur, Lüfter; realisiert auf RM400 C90 unter Reliant Unix)

@ Filesystemauslastung der höchstbelasteten Partitionen

Aufgrund der in der Entwicklungsumgebung des Prototypen verfügbaren Serversysteme wurden alle Managementvorgänge zunächst unter Reliant Unix entwickelt. Mit wenigen Ausnahmen lassen sich ohne Veränderungen auch Solaris-, Linux- und sonstige Unix-Server überwachen.

Es hat sich gezeigt, dass ein professionelles Systemmanagement auch ohne Installation proprietärer Clientsoftware realisierbar ist. Die Auslegung der grafischen Oberfläche als HTML-Seite ermöglicht eine neue Dimension der Flexibilität. Die fortlaufend aktualisierte Visualisierung der kritischen Betriebsparameter können in beliebige Webseiten integriert werden. Größe und Aktualisierungsintervall sind frei wählbar. Eine entsprechende Freischaltung per Webserver und Firewall vorausgesetzt, ist eine Überwachung über beliebige IP-basierte WANs möglich. Dies ist insbesondere für die Themenbereiche Bereitschaft und Outsourcing interessant.

Bei eingehenden Problemmeldungen hilft das System bei der Zuordnung des Problems auf die betroffenen Server. Lastsituationen lassen sich schnell erfassen; Routinevorgänge bei der Systemüberwachung sind nun auf einen Blick möglich.

01.04 Missverständnisse, Unterlassungssünden und Denkfehler

Die Defizite im KapMan-Prozess sind nicht ausschließlich technischer sondern oft organisatorischer und struktureller Art.

Die folgende Übersicht ist eine Zusammenstellung, die aus den praktischen Erfahrungen vieler Jahre entstanden ist. Sie ist in der Grobgliederung systematisch, erhebt aber keinen Anspruch auf Vollständigkeit.

(a) Problemkreis 1: Hardware-zentrierte Sichtweise

@ "Performance-Engpässe können durch Hardwareerweiterungen beseitigt werden." Das ist zwar oft richtig, aber eben nicht immer. Es gibt ausreichend Erfahrungsberichte aus der Praxis, dass Erweiterungen der Hardware nicht zu der angestrebten Performanceverbesserung geführt haben. Hier einige Beispiele:

- CPU-Upgrade bei Engpässen im Speicher oder I/O-Bereich;
- Erhöhung der Prozessoranzahl bei schlecht skalierenden Architekturen bzw. Applikationen;

- Erhöhung der Übertragungsbandbreite bei ineffizienter Protokollsoftware, z.B. muss eine LAN-fähige Applikation bei "Stop-and-wait"-Verhalten auf WAN-Strecken trotz ausreichender Bandbreite nicht notwendig performant laufen.

@ Das Thema Kapazitätsmanagement wird zu hardwarezentriert gesehen. Softwarebottlenecks werden nicht erkannt bzw. außerhalb des eigenen Verantwortungsbereichs gesehen, d.h. bei Softwareentwicklern und Softwaredistributoren, welche aus Kosten- und Termindruck unperformante Software ausliefern. Bei der Softwareentwicklung wird Software Performance Engineering meist nicht einmal ansatzweise betrieben. So steigt bekanntlich der Ressourcenbedarf beim Einsatz neuer Software-Releases deutlich an. Es wird erwartet, dass der Preisverfall im Hardwarebereich die Softwareengpässe kompensiert.

@ Laststeigerung kann oft nicht durch Hinzufügen von "steckbarer Leistung" abgefangen werden. Die Skalierfähigkeit von großen Systemen ist wegen der Nicht-Linearität des Systemverhaltens schlecht einschätzbar. Ein typisches Beispiel ist der Anstieg von Antwortzeiten, der im Niedriglastbereich linear, aber im Hochlastbereich exponentiell erfolgt. Z.B. kann eine Lastverdoppelung (+100 %) ohne spürbare Performance-Einbuße bleiben, weitere 20 % gehen mit erheblicher Dienstgütebeeinträchtigung einher.

@ Die Entwicklung im Bereich der Hardware vollzieht sich so rasant, dass die Versuchung groß ist, fast jedes unbefriedigende Systemverhalten durch erhöhte CPU-Leistung, leistungsfähigere Peripherie, erhöhte Netzbandbreiten etc. abzustellen. Bei sinkenden Hardware-Preisen sind Tendenzen erkennbar, dass die Anwender die verlockenden Angebote der Hardware-Anbieter annehmen und ihre Systeme austauschen. Grundlegende Designprobleme lassen sich aber durch schnellere Systeme nicht lösen. Vielmehr führt die Aufrüstung der Hardware in vielen Fällen dazu, dass die Systemauslastung weiter sinkt, ohne dass sich dieses in verbesserten Antwortzeiten für den Anwender niederschlägt (Software-Engpass).

@ Im Netzwerkbereich ist man häufig froh, "... dass es funktioniert": Im Design werden nicht selten nur die unteren Ebenen betrachtet, und die Anforderung, Applikationen auf Ebene 7 performant und zuverlässig zu betreiben, bleibt dann unberücksichtigt

@ Schlagworte wie "Storage on Demand" suggerieren grenzenlose Ressourcenverfügbarkeit.

(b) Problemkreis 2: Organisation und Management

@ Berührungsängste, Konkurrenzdenken und Kompetenzkämpfe der Fachabteilungen verhindern eine ganzheitliche Sicht. Oft werden statt des Gesamtsystems nur Einzelaspekte betrachtet, z.B. wird der Blick nur auf das Netz oder nur auf den Datenbank-

Server gerichtet. Ein Beispiel für einen Dialog aus der Praxis: „Wie ist der Server ausgelastet?“ – „Wir haben meist 1 GByte als Reserve!“ – „Nein, ich meine nicht die Hauptspeicherauslastung, sondern wie hoch ist die CPU-Auslastung?“ – „Das weiß ich nicht, dafür ist mein Kollege zuständig.“

@ Falsch ist: "Kapazitätsmanagement ist eine technische Aufgabe, um welche sich die Betreuer von Hosts, Servern und Netzwerken zu kümmern haben." Kapazitätsmanagement ist eng mit den Geschäfts- und Unternehmenszielen verbunden. Die Unternehmensleitung muss daher entsprechende personelle Ressourcen bereitstellen.

@ "Die Kapazitätsmanagementeinführung ohne unternehmensstrategische Basis kostet nur Geld. In anderen Worten: Methoden und Tools und ein Mann, der nebenbei damit arbeitet, macht noch lange kein erfolgreiches Kapazitätsmanagement" (Zitat: Jürgen Beust, CMG-CE-Jahrestagung, Bremen 1994.)

@ Die Anforderungen des Tagesgeschäfts lassen keine Zeit für Kapazitätsmanagement. Prophylaktisches Denken und proaktives Handeln wird von der kurzfristigen Dynamik dominiert bzw. verdrängt. Gegenparole: "Stop fire fighting, start planning!".

@ Der Aspekt des „Total Cost of Ownership“ wird oft übersehen. Z. B. werden die Planungskosten für IT-Systeme (für Kapazitätsmanagement, Monitoring, Prognose) häufig zu niedrig angesetzt. Den wenigen Prozent (oft auch < 1%) der Gesamtkosten können immense Folgekosten aufgrund von verfehlten Zielen gegenüber stehen.

@ Die Erwartungshaltung der KapMan Consultants sollte nicht zu hoch sein. Unsere Erfahrungen zeigen, dass es deutlich leichter ist, über die Beseitigung von Tagesproblemen (Tuning) als über Planung in das Consulting einzusteigen. In diesem Fall ist die Akzeptanz des Anwenders für das Thema Kapazitätsmanagement meist schon vorhanden.

(c) Problemkreis 3: Messung und Monitoring des IST-Zustands

@ Eine mangelnde Kenntnis vom Zustand des gesamten IT-Systems bzgl. benutzer- und betreiber-orientierten Maßen ist eine für viele Installationen typische Situation. Konkret: Betreiber haben keinen Überblick über die Ressourcenauslastungen. Benutzer sind meist nicht in der Lage, konkrete Aussagen über die (meist implizit definierten) Dienstgüten zu machen.

@ Die Zuordnung und Synchronisierung von Messergebnissen aus unterschiedlichen Quellen ist problematisch. Meist sind die Zeitintervalle für die Messungen nicht ausreichend überlappend. Typisches Beispiel dafür: „Heute mess´ ich die Prozessorauslastung, morgen das I/O-System und übermorgen die Antwortzeiten.“

- Messungen sollten zielgerichtet erfolgen, d.h. nicht alles muss gemessen werden, insbesondere, wenn dadurch zusätzlicher Overhead verursacht wird, welcher den Betrieb behindert. In Kürze formuliert: "Wer viel misst, misst Mist!"
- Die Abbildung logischer Prozesse auf physikalische Ressourcen wird ständig komplexer und ist nicht eindeutig bzgl. Leistungsabrechnung.
- Die Vergabe von Messaufträgen an eine objektive und kompetente Instanz ist wünschenswert, um sicher zu stellen, dass die Vorgaben bzgl. Umfang, Messintervall und Messobjekten eingehalten werden.

(d) Problemkreis 4: Planung, Prognose und Werkzeuge

- Ein gutes Werkzeug kann Probleme lösen, aber das Know-how von kompetenten Personen ist noch wichtiger. Konkret: Software-Lizenzen für Werkzeuge sind zwar teuer, aber die Kosten für den Personalaufwand sind noch teurer (weshalb oft an zweitem gespart wird).
- Modellierung kann billiger als Messung und Monitoring sein. Aber: Die durch Messung und Monitoring erhaltenen Daten sind Voraussetzung für eine sinnvolle Modellierung.
- Der Detailliertheitsgrad der Modelle steht nicht in Einklang mit den verfügbaren Messdaten und den Zielen der Performanceanalyse. Insbesondere bei Simulationsmodellen kann das ein Problem werden. Hierfür sind detaillierte Messdaten über Bedien- und Wartezeiten erforderlich, wobei Mittelwerte nicht ausreichen. Es wird dann eine “Genauigkeit“ der Simulationsergebnisse suggeriert, der die Messergebnisse für die Modellierungsparameter nicht gerecht werden können.

KAPITEL 02 ANFORDERUNGEN AN DAS KAPAZITÄTS-MANAGEMENT

R. BORDEWISCH, C. FLUES, R. GRABAU, J. HINTELMANN,
K. HIRSCH, H. RISTHAUS

02.01 Kapazitätsmanagement: Ziele und Inhalte

Im Folgenden wird skizziert, was Kapazitätsmanagement (KM) umfasst und welche Bedeutung MAPKIT als Voraussetzung für ein erfolgreiches Kapazitätsmanagement besitzt.

(a) Kapazitätsmanagement

Kapazitätsmanagement beinhaltet alle Konzepte, Aktivitäten und Abläufe, die darauf abzielen, die in einem Unternehmen zur Verfügung stehenden IT-Einrichtungen unter gegebenen Randbedingungen bestmöglich zu nutzen und die in Zukunft erforderlichen Kapazitäten von IT-Komponenten bedarfsgerecht und wirtschaftlich zu planen.

Kapazitätsmanagement ist eine Basis für eine erfolgreiche Systemintegration und verfolgt folgenden methodischen Ansatz:

- Identifizieren der Geschäftsprozesse des Unternehmens
- Beschreiben der Geschäftsprozess-Performance-Ziele
- Abbilden von Geschäftsprozessen auf die erforderlichen IT-Prozesse
- Definieren der relevanten Anforderungen der Anwender
- Planen und bestimmen der benötigten IT-Komponenten
- Überwachen und steuern des laufenden Betriebs
- Analyse und Tuning

Dieser Ansatz zielt darauf ab, durch geeignete Beschreibung und Bewertung der verschiedenen Einzelbestandteile aus einer ganzheitlichen Sicht das Gesamtsystemverhalten zu verbessern.

Mit Blick auf die klassischen Projektphasen ist Kapazitätsmanagement im Idealfall ein permanenter Prozess in den Phasen:

- Planung
- Ausschreibung
- Angebot
- Realisierung
- Abnahme
- Betrieb

Kapazitätsmanagement trägt wesentlich zur Erreichung folgender Unternehmensziele bei:

- Planungs- und Investitionssicherheit,
- Gesicherter IT-Betrieb und optimale Ressourcen-Nutzung,
- Verbesserte Wirtschaftlichkeit bedingt durch Kostenreduktion des IT-Betriebs und durch Verbesserung des IT-Leistungsangebotes.

Daneben kann es als Basis für eine Kostenzuordnung nach dem Verursacherprinzip dienen (Kostentransparenz). Das führt schließlich zur Produktivitätssteigerung und zur Verbesserung der Wettbewerbsfähigkeit.

(b) Methodik des Kapazitätsmanagements

Das Kapazitätsmanagement ist im Mainframe-Bereich seit langem üblich und daher eine systematisch ausgearbeitete und bewährte Disziplin. In verteilten, heterogenen Systemen sind diese Aufgaben wesentlich schwieriger zu erfüllen als in einer homogenen und räumlich konzentrierten Mainframe-Landschaft. Mit MAPKIT wird eine problemadäquate und fortschreibbare Methodik für das Kapazitätsmanagement von verteilten und kooperierend arbeitenden Systemen entwickelt, erprobt und etabliert. Unterstützt wird die Methodik durch die Bereitstellung entsprechender Verfahren und Werkzeuge. Die praktische Umsetzung der Methodik ermöglicht die effiziente Durchführung von Kapazitätsmanagement. Auf diese Weise wird es möglich, neuen Herausforderungen auf dem Gebiet der Systemplanung durch Anwendung und Weiterentwicklung der gewonnenen Erkenntnisse systematisch zu begegnen.

Im Folgenden wird erläutert, welchen Anforderungen ein modernes Kapazitätsmanagement hinsichtlich der Bewertungskriterien Verfügbarkeit, Zuverlässigkeit und Performance Rechnung tragen muss, um heterogene IT-Systeme wirtschaftlich planen und betreiben zu können.

02.02 Generelle Anforderungen an das Kapazitätsmanagement

In diesem Abschnitt werden allgemeine Anforderungen an die Methodik des Kapazitätsmanagements in heterogenen, verteilten IT-Landschaften vorgestellt.

(a) Sensibilisierung der Auftraggeber für Kapazitätsmanagement

IT-Landschaften werden zunehmend komplexer und müssen gerade im Zeitalter des Internets immer vielfältigeren Ansprüchen genügen. Darüber hinaus unterliegen sie einem Strukturwandel. Zentrale, umlagenfinanzierte Strukturen innerhalb eines Unternehmens werden abgelöst durch selbständige IT-Bereiche. Vielfach werden durch Outsourcing ganze IT-Bereiche zu neuen Provider-Firmen, die an einer verursachergerechten Abrechnung ihrer erbrachten Leistung interessiert sind. Auf der anderen Seite verlangen die Anwender dieser Provider eine genaue und beweisbare Aufstellung der ihnen in Rechnung gestellten Leistung.

Der Umgang mit dieser Situation erfordert strategisches und systematisches Denken und Handeln. An erster Stelle muss bei Managern und Planern das Bewusstsein geschaffen werden, dass

- nur effiziente IT-Prozesse eine adäquate Unterstützung der Geschäftsprozesse (GP) eines Unternehmens ermöglichen.
- das Verhalten der beteiligten IT-Komponenten und das Anwenderverhalten miteinander in Wechselwirkung stehen.
- es sich bei IT-Systemen um lebendige Systeme mit wechselnden Anforderungen handelt, die im laufenden Betrieb permanent anhand gültiger Kriterien überwacht, analysiert und bewertet und die für zukünftige Anforderungen geplant werden müssen.
- die Aufrüstung mit schnelleren Rechnern oder mehr Netzbandbreite die prinzipiellen Architektur- und Designprobleme komplexer Anwendungen nicht dauerhaft löst.
- Kapazitätsmanagement ein wesentlicher Bestandteil der Systemplanung ist.

(b) Existenz von Dienstgüte-Vereinbarungen (SLAs)

Eine Grundvoraussetzung zur erfolgreichen Durchführung von Kapazitätsmanagement ist die Existenz von Dienstgütevereinbarungen (Service Level Agreements). Diese Vereinbarungen sind auf unterschiedlichen Ebenen erforderlich. Dabei kommt der Erfassung der Geschäftsprozesse eine zentrale Bedeutung zu. Sie liefern einen Beitrag zur Wertschöpfungskette des Unternehmens, d.h. der Nutzen, der aus ihnen erwächst, bestimmt indirekt den Aufwand, den Kapazitätsmanagement kosten darf.

Eng verknüpft damit ist die adäquate Beschreibung der IT-Prozesse. Adäquat heißt hier, dass sowohl zeitliche als auch räumliche Beziehungen zu den Geschäftsprozessen erkennbar sein müssen. IT-Prozesse erbringen IT-Dienste zur Unterstützung der Geschäftsprozesse. Die Dienste erbringen sie mit Hilfe von IT-Komponenten. Durch dedizierte Zuordnungen zwischen den Ebenen „Geschäftsprozesse“, „IT-Prozesse“ und „IT-Komponenten“ können Messergebnisse auf Ebene der IT-Komponenten in Ergebnisse für die IT-Prozesse und letztlich für die Geschäftsprozesse transformiert werden. Andererseits lassen sich Dienstgüteanforderungen auf Geschäftsprozess-Ebene (GP-Ebene) entsprechend auf die IT-Prozess-Ebene und wo dies nötig ist, auch auf die IT-Komponenten übertragen.

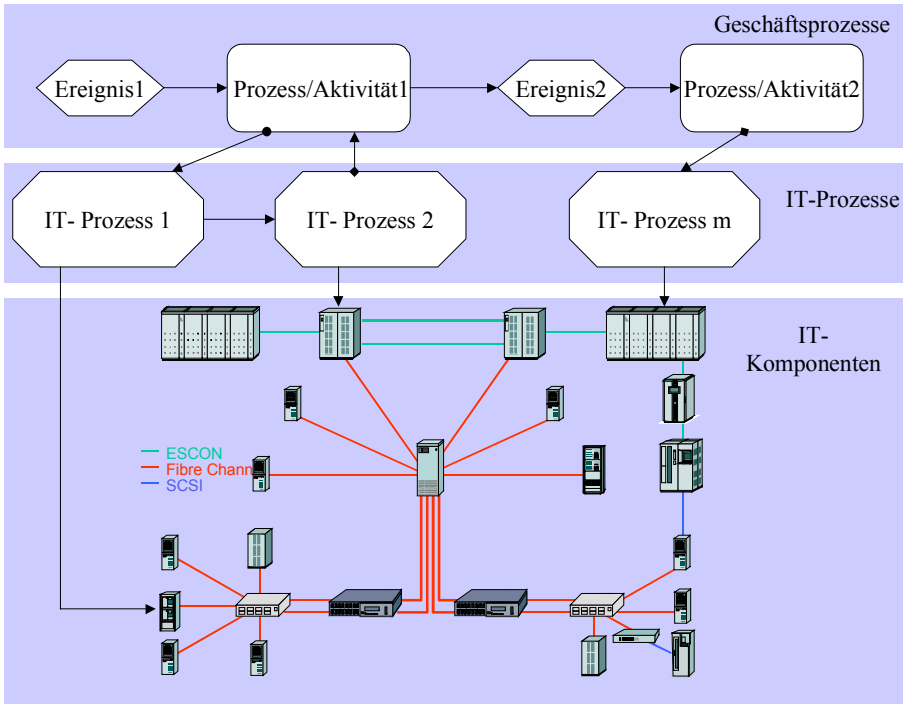


Abbildung 02-1: Systembeschreibung im Kapazitätsplanungsprozess

Unabhängig von der Ebene lassen sich grundlegende Bedingungen für SLAs formulieren. Die Spezifikation der Dienstgüte muss „SMART“ sein, d.h.

- Specific: Quantitative Angaben statt „besser“, „schneller“ etc.
- Measurable: Vereinbarte Werte müssen messtechnisch erfassbar und im laufenden Betrieb überprüfbar sein.

@ Acceptable: Vereinbarte Werte müssen anspruchsvoll und von allen akzeptiert sein.

@ Realizable: Vereinbarte Werte müssen erreichbar sein, sie dürfen nicht überzogen sein.

@ Thorough: Dienstgütevereinbarungen müssen gründlich und vollständig erfolgen.

Zu jeder Dienstgütevereinbarung gehört eine Lastvereinbarung. Damit verpflichtet sich der IT-Anwender zur Einhaltung der festgelegten Belastung. Sind ausreichende Reservekapazitäten vorhanden, können kurzfristige Überschreitungen zugelassen werden. In diesem Fall sind entweder erhöhte Kosten zu tragen oder es ist eine Absenkung von Performance-Werten zu dulden.

SLAs werden für die Kategorien

@ Verfügbarkeit

@ Zuverlässigkeit

@ Performance

geschlossen. Wenn die Definition von Performance-Maßen nicht einheitlich ist, müssen gegebenenfalls die Messpunkte und Messverfahren mit in die Vereinbarung aufgenommen werden. Die SLAs legen die Zielgrößen für die Planung neuer und die Überwachung existierender IT-Systeme fest.

(c) Methode zur plattformübergreifenden Beschreibung und Analyse des Gesamtsystemverhaltens

Um ein Gesamtsystem plattformübergreifend beschreiben und analysieren zu können, sind verschiedene Betrachtungsebenen erforderlich, die sich stark von technischen Gegebenheiten lösen. Daher wird ein Ansatz gewählt, der sich unabhängig von der IT-Architektur, wie z.B. Mainframe-Orientierung oder Client-Server-Architektur, an den Zielen und Prozessen eines Unternehmens orientiert.

Zentrales Anliegen von IT-Betreibern und IT-Anwendern ist die tägliche, effiziente Erfüllung aller Anforderungen, unabhängig von der Komplexität der vorhandenen IT-Landschaft. Dabei ergeben sich die Aufgabenschwerpunkte:

@ Überwachung, Analyse und Tuning

@ Planung und Prognose

Überwachung, Analyse und Tuning

Für den laufenden Geschäftsbetrieb ist es notwendig, das Verhalten der beteiligten Systeme/Systemkomponenten durch ein umfassendes Monitoring überwachen zu können.

Die zum System gehörenden Komponenten werden hinsichtlich

- Verfügbarkeit
- Zuverlässigkeit
- Auslastung, Verweilzeiten, Funktionsaufrufe

anhand der vereinbarten Dienstgüten überwacht.

Die Anwendungen werden hinsichtlich

- Antwortzeitverhalten
- Durchsatz
- Ressourcenbedarf und
- Auftrittshäufigkeiten

überwacht.

Neben der Sicherstellung der Betriebsbereitschaft werden Komponenten für ein proaktives Management benötigt, d.h. Komponenten, die das Auftreten von Engpässen so frühzeitig erkennen, dass es zu keiner Beeinträchtigung der Leistungsfähigkeit kommt.

Darüber hinaus werden Komponenten für eine Bewertung und Abrechnung (Accounting) der in Anspruch genommenen Leistung benötigt. Die Ermittlung muss auf IT-Prozess- und Geschäftsprozess-Ebene erfolgen können.

In diesem Zusammenhang sind auch die Definition und die Überwachung von Alarmwerten bzw. die Definition von Reaktionen bei Erreichung bestimmter Schwellwerte zu betrachten. Die Schwellwerte stellen aus technischer Sicht die messbaren Kriterien der für den ordnungsgemäßen Betrieb vereinbarten Dienstgüten (Service Level Agreements) dar. Eine Über- bzw. Unterschreitung der bestimmten Schwellwerte kennzeichnet damit die Verletzung einer Vereinbarung.

Die Norm DIN 66273 (Messen und Bewerten der DV-Leistung) bietet einen Bewertungsrahmen hinsichtlich Einhaltung von Leistungsanforderungen, insbesondere für die Termintreue bei der Auftragserfüllung.

Darüber hinaus müssen Fehler und wiederholt auftretende Engpässe im Gesamtsystem im Rahmen von Tuning-Maßnahmen möglichst schnell und kostenoptimiert beseitigt werden, um die Effizienz des Gesamtprozesses nicht nachhaltig zu gefährden.

Planung und Prognose

Zur Planung und Prognose gehören folgende Teilaufgaben:

@ Die Beschreibung der Last und Dienstgüte-Anforderungen.

@ Die Voraussage von Sättigungszeitpunkten des unveränderten Systems bei Annahme von Lastevolution.

@ Die Beschreibung und Bewertung von Änderungen des organisatorischen oder technischen Umfeldes:

- Plattformwechsel, Versionswechsel
- neue oder zusätzliche Applikationen und Dienste
- neue Topologien
- erhöhtes Lastaufkommen (mehr Anwender)
- veränderte Geschäfts- und IT-Prozesse

(d) Ermittlung der Anforderungen des IT-Betriebs

In diesem Abschnitt werden die allgemeinen Anforderungen aus Sicht des IT-Betriebs an die Methodik des Kapazitätsmanagements beschrieben. Bei der Formulierung der Anforderungen wird allgemein zwischen einem Dienstnehmer (Service User) und einem Diensterbringer (Service Provider) unterschieden. Weiterhin kann differenziert werden in Unternehmensleitung, IT-Leitung, IT-Fachbereiche und Anwender-Fachbereiche. Die Anforderungen an diese Personengruppen können in Abhängigkeit der Zugehörigkeit zu einem Dienstnehmer oder Diensterbringer variieren.

Unternehmensleitung

Die Unternehmensleitung definiert die Unternehmensziele und stellt die finanziellen Mittel zur Verfügung, die zur Erreichung der Ziele notwendig sind. Sie muss davon überzeugt sein oder werden, dass Kapazitätsmanagement ein unverzichtbarer Baustein für ein erfolgreiches Unternehmen ist.

Die Unternehmensleitung muss rechtzeitig und möglichst transparent die erforderlichen Investitionen erkennen und disponieren können. Das bedeutet, dass Kapazitätsmanagement

der Unternehmensleitung geeignete Informationen, den Planungszyklen entsprechend, bereit stellen muss, um so dazu beizutragen, die Aufgaben besser und zuverlässiger erfüllen zu können. Zu den erforderlichen Informationen zählen Anschaffungs- und Installationskosten für Server, Clients, Netzkomponenten, etc. aber auch Software-Updates oder Neuanschaffungen. Darüber hinaus kann es notwendig sein, Mess- und Monitoring-Komponenten zu installieren.

IT-Leitung

Sie dient als Schaltstelle und Vermittlung zwischen Fachbereichen und Unternehmensleitung und sorgt so für die technische Unterstützung der Geschäftsprozesse.

IT-Fachbereiche

Hier sind die Systemadministratoren und IT-Fachleute gemeint, die für den reibungslosen Ablauf des IT-Betriebs inkl. aller Anwendungen sorgen.

Anwender-Fachbereiche

In diese Gruppe fallen die Sachbearbeiter, die die IT-Dienstleistung in Anspruch nehmen und deren Management, das die Geschäftsprozesse und die Anforderungen der Endanwender formuliert. Je nach Organisationsform der Unternehmen können Gruppen zusammen gefasst sein oder ganz entfallen.

Anforderungen der Service-User

Das primäre Interesse der Service-User ist die Ausführung ihrer Geschäftsprozesse unter akzeptablen Randbedingungen. Sie formulieren die konkreten Anforderungen an das System in Form von Dienstgüteparametern für alle Kategorien. Für Sachbearbeiter sind die wesentlichen Anforderungen kurze, der Komplexität der Aufgabe entsprechende Antwortzeiten und hohe Verfügbarkeit.

Die Überprüfung der ausgehandelten Werte ist Bestandteil des Service-Managements und muss durch ein entsprechendes Mess- und Monitoring-Konzept unterstützt werden. Für den Anwender ist es wesentlich, dass dazu Werte erfasst werden, die seine Anforderungen zutreffend beschreiben. Insbesondere sollen die Messwerte benutzt werden können, um vereinbarte Dienstgüteparameter zu überprüfen.

Anforderungen der Service-Provider

Um kostengünstig und wettbewerbsfähig handeln zu können, ist die ständige Überwachung des laufenden Betriebs unter technischen und wirtschaftlichen Gesichtspunkten notwendig. Dies gilt insbesondere dann, wenn der Diensterbringer nicht als Stabsbereich im Unter-

nehmen eingegliedert ist und sich aus Umlagen heraus finanziert. Selbständige oder als Profit Center organisierte IT-Bereiche haben ein lebenswichtiges Interesse daran, die von ihnen erbrachten Leistungen verursachergerecht abrechnen zu können. Dabei ist es aus Wirtschaftlichkeitsaspekten von Interesse, welcher Kostenaufwand durch Nichteinhaltung von Dienstgütevereinbarungen entsteht (z.B. bei fehlender Termintreue, wie sie in DIN 66273 definiert ist).

Für eine vorausschauende und langfristige Planung des Betriebs ist es ferner notwendig, eine über kurze Zeiträume hinausgehende Sichtweise zu entwickeln. Auf diese Weise werden Investitionen langfristig planbar. Außerdem liefert diese Sichtweise Argumentationshilfe bei der Planung und Rechtfertigung des IT-Budgets gegenüber dem Management. Dazu gehören auch die Analyse und Prognose der Bedarfsentwicklung im Netz- und Server-Bereich hinsichtlich der Übertragungsbandbreiten, der Rechenleistung, des Speicherbedarfs etc.

Service-Provider benötigen ein umfassendes Mess- und Monitoring-Konzept, mit dessen Hilfe sie einzelne Komponenten in ihrem System bewerten können. Hierzu zählen die Auslastung von Systemkomponenten, wie CPU, Speicher, Netzelemente, aber auch die Zugriffszeiten auf Datenbanken, Durchlaufzeiten von Paketen in Netzen etc. Diese punktuellen Messwerte alleine helfen lediglich bei einer Engpassanalyse bzw. bei der Beantwortung der Frage, ob die Komponenten eine ausreichende Leistungsfähigkeit besitzen. Für ein integriertes übergreifendes Management ist es wichtig, die Messdaten einzelnen IT-Prozessen und letztlich den Geschäftsprozessen zuzuordnen. Dabei existiert immer eine n-zu-m-Beziehung zwischen Geschäftsprozessen und IT-Prozessen. Die Zuordnung von IT-Verbräuchen zu den existierenden IT-Prozessen sollte ebenfalls möglich sein. Darüber hinaus muss der Service-Provider auch in der Lage sein, Anwendungen in ihren Wegen durch ein verteiltes System von Einzelkomponenten verfolgen und Teilstrecken vermessen zu können.

(e) Zielkonflikte und Gemeinsamkeiten

Zielkonflikt: Antwortzeit versus Auslastung

Bei Einhaltung der Anforderungen der Service-User, z.B. hinsichtlich kurzer Antwortzeiten, ist mit einer Steigerung ihrer Produktivität zu rechnen. Im anderen Fall sind durch lange Wartezeiten oder bei Problemen mit der Verfügbarkeit von Anwendungen Frustreaktionen der Benutzer zu erwarten, die die Performance der betroffenen Geschäftsprozesse beeinträchtigen. Der Service-Provider ist demgegenüber an einer möglichst hohen Auslastung der verwendeten Komponenten interessiert, um eine wirtschaftliche Nutzung zu gewährleisten.

Beide Ziele sind i.a. nicht gleichzeitig erreichbar, da kurze Antwortzeiten hohen Auslastungsgraden widersprechen. Sollte die Umsetzung einzelner Anforderungen durchführbar, aber aus Kostengründen nicht sinnvoll sein, müssen Kompromisse gefunden werden.

Gemeinsamkeiten: Dienstgüte und Kostentransparenz

In vielen Fällen stimmen die Anforderungen von Service-Provider und Service-User überein, denn beide sind an einer gerechten Bewertung von Leistung interessiert und wollen ein möglichst günstiges Preis/Leistungsverhältnis erreichen. Für eine optimale Zusammenarbeit sind genaue Definitionen der Schnittstellen zwischen Kunden, Lieferanten und Partnern erforderlich.

KAPITEL 03 METHODIK UND VERFAHREN

R. BORDEWISCH, C. FLUES, R. GRABAU, J. HINTELMANN,
K. HIRSCH, F.-J. STEWING

03.01 MAPKIT-Vorgehensmodell

Grundvoraussetzung für das Gelingen eines KapMan-Vorhabens ist die Bestimmung und Schaffung der notwendigen Rahmenbedingungen. Die Erfahrungen haben gezeigt, dass die anscheinend natürliche Verankerung von KapMan in dem Vertriebsprozess aufgrund der oft gegensätzlichen Interessen nicht praktikabel ist. Vielmehr ist der KapMan-Prozess als eigenständiger Serviceprozess im System Life Cycle zu sehen. Sind die notwendigen Rahmenbedingungen geschaffen, gibt es für das weitere Vorgehen zwei Alternativen: Der Einstieg über eine Überwachung des existierenden IT-Systems oder über die Planungs- und Prognosephase.

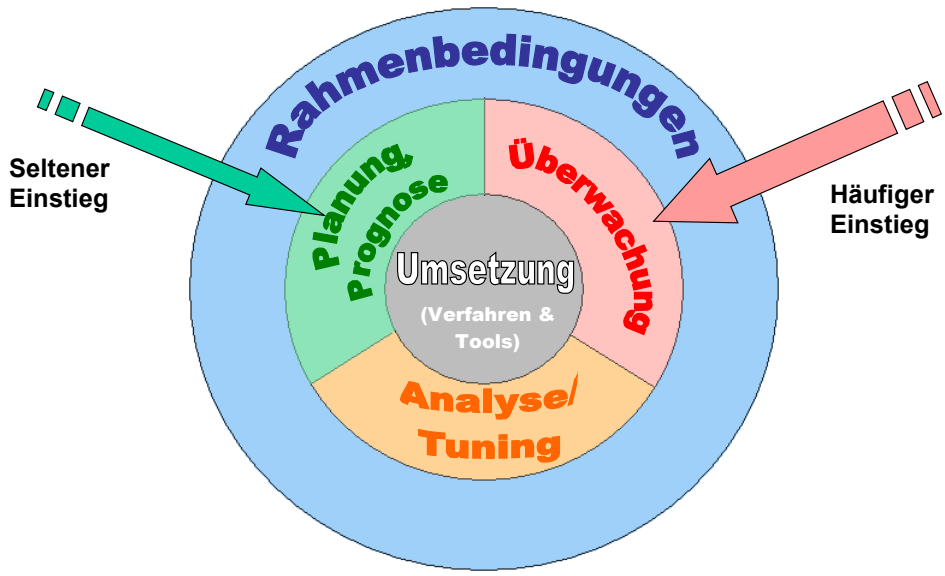


Abbildung 03-1: MAPKIT-Vorgehensmodell

Der Einstieg über die Überwachung des existierenden Systems, in dem oft bereits Performance-Probleme auftreten, ist in der Praxis am häufigsten realisierbar. Hierbei wird das IT-System des Kunden zunächst vermessen und analysiert. Die Erkenntnisse münden i. d. R. in einem Tuning des existierenden Systems, um die Performance-Probleme zunächst kurz- bzw. mittelfristig zu lösen.

Anschließend sollte eine Planungs- und Prognosephase folgen, um die Systemkapazitäten langfristig bedarfsgerecht zu planen. Die daraus gewonnenen Erkenntnisse sollten, wenn nötig, zu einer Modifikation des Systems führen, damit Performance-Probleme im laufenden Betrieb möglichst gar nicht erst auftreten.

An die Überwachung des existierenden Systems kann sich alternativ direkt die Planungs- und Prognosephase anschließen, falls ein Tuning des bestehenden Systems nicht nötig ist, weil keine Performance-Probleme existieren.

Die Auswirkungen der sich aus der Planungs- und Prognosephase ergebenden Systemmodifikation sollten wiederum überwacht werden, um über einen Soll/Ist-Vergleich ein Feedback über die Prognose zu erhalten und um ggf. weitere Maßnahmen einzuleiten. Damit schließt sich der Kreis. Der Zyklus kann erneut beginnen.

Ein in der Praxis seltener anzutreffender, aber durchaus vernünftiger Einstieg erfolgt über die Planungs- und Prognosephase. Es werden zunächst Ziele und Anforderungen für das zukünftige System definiert, ohne das existierende System detailliert zu analysieren. Dieser Weg ist sinnvoll, wenn z.B. das zukünftige System mit dem existierenden wenig Ähnlichkeit aufweist. Abbildung 03-2 verdeutlicht den Ablauf der einzelnen Phasen des Vorgehensmodells. Auf die Darstellung der Phase „Festlegung der Rahmenbedingungen“ wurde aus Gründen der Übersichtlichkeit verzichtet. Diese Phase stellt beim Beginn des Kapazitätsmanagements den zentralen Punkt dar. Die Rahmenbedingungen müssen aber darüber hinaus während des ganzen Managementprozesses überwacht bzw. modifiziert werden, damit das Kapazitätsmanagement zum Erfolg führen kann.

Die Umsetzung des MAPKIT-Vorgehensmodells erfordert Methoden und Werkzeuge, die die Aktivitäten in den einzelnen Phasen unterstützen bzw. überhaupt erst möglich machen.

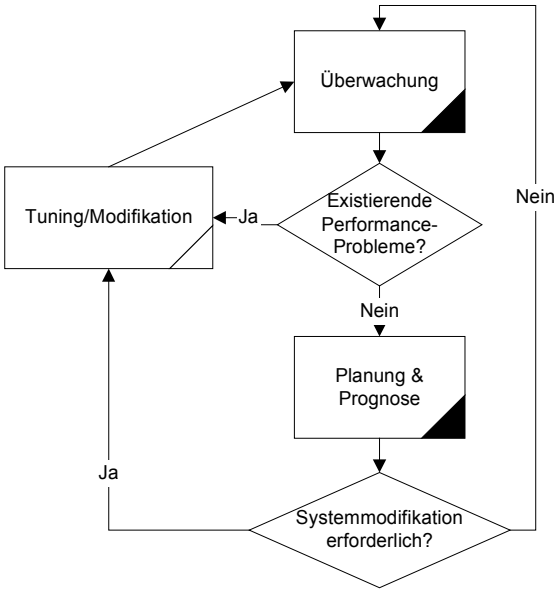


Abbildung 03-2: Zusammenspiel der Phasen des Vorgehensmodells

Eine wesentliche Erkenntnis ist, dass heute relativ wenig Aufwand (und damit Kosten) in die Phase "Planung und Prognose" investiert wird.

Damit ist der KapMan-Einstieg über die Phase "Planung und Prognose" zwar erstrebenswert, aber auch sehr schwierig.

Häufig wird KapMan dann erforderlich und angewendet, wenn "das Kind schon in den Brunnen gefallen ist". Bei akuten Performance- und Verfügbarkeitsproblemen werden dann Detail- und Ursachenanalysen durchgeführt, um durch Tuning das Systemverhalten zu verbessern. Somit ist der KapMan-Einstieg in dieser Phase relativ leicht und einfach.

In den letzten Jahren hat man erkannt, dass die permanente Überwachung des Systemverhaltens erforderlich ist, um mögliche Engpässe rechtzeitig erkennen zu können, um ein Gegensteuern zu ermöglichen. Aus der Ermittlung und Überwachung der Dienstgüte und der Auslastung der Systemressourcen werden Trends abgeleitet, die den KapMan-Einstieg in der Phase „Planung und Prognose" ermöglichen.

Ziel eines umfassenden Kapazitätsmanagements muss es sein, wesentlich mehr als heute für Planung und Prognose zu investieren und das aufwändige Analysieren und Tunen zu reduzieren, um letztendlich eine Ausgewogenheit von "Planung und Prognose", "Überwachung" und "Analyse und Tuning" zu erreichen (vgl.

Abbildung 03-3).

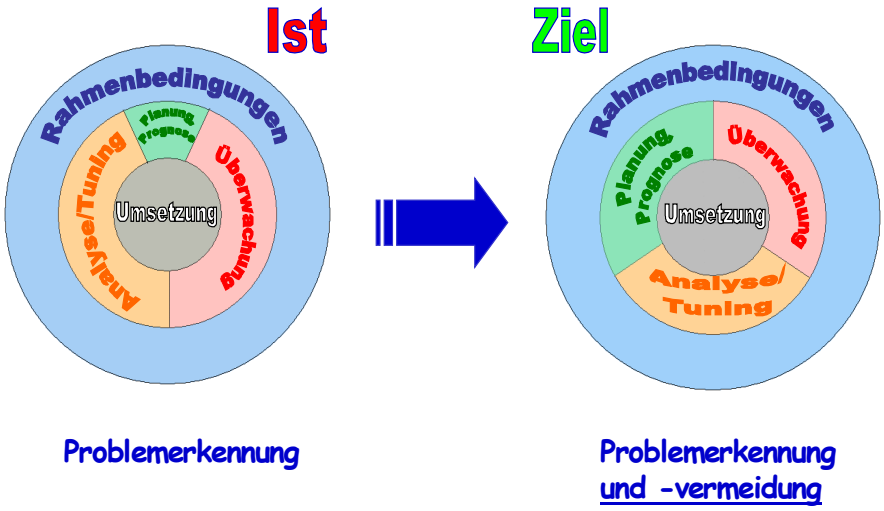


Abbildung 03-3: IST- und SOLL-Zustand

(a) Festlegung der Rahmenbedingungen

Zur Anwendung und Etablierung der KapMan-Methodik müssen Vorarbeiten geleistet werden.

Allgemeine Rahmenbedingungen

Die Abklärung der allgemeinen Rahmenbedingungen, innerhalb derer das Kapazitätsmanagement durchgeführt werden kann, schafft eine Arbeitsbasis für alle Beteiligten. Diese Rahmenbedingungen umfassen unternehmenspolitische, organisatorische und finanzielle sowie technische Aspekte. Deshalb muss das IT-Management unbedingt miteinbezogen

werden, und zur Festlegung der Service-Güte müssen die Fachabteilungen und Endbenutzer ebenso involviert sein wie die IT-Fachleute zur Spezifizierung der IT-Prozesse.

Durch die vorgegebenen Bedingungen kann der Einsatz von Verfahren oder Werkzeugen eingeschränkt werden. Im ungünstigsten Fall lassen sich Ziele des Kapazitätsmanagements unter den gegebenen Rahmenbedingungen nicht oder nur unvollständig erreichen. Dies muss dem Anbieter und Anwender vorab verdeutlicht werden. Nur wenn das IT-Management, die Fachabteilungen bzw. die Endbenutzer und die IT-Fachleute hinsichtlich der Anwendung und des Nutzens von KapMan sensibilisiert sind, kann das KapMan-Vorhaben gelingen. Zur Erreichung dieses Zieles und zur Bearbeitung der folgenden Punkte bietet sich ein einführender Workshop an. Zur Erleichterung der Erfassung der benötigten Eingangsdaten (Unternehmensziele, Geschäftsprozesse) ist es hilfreich, verfügbare Fragebögen und Checklisten als Hilfsmittel einzusetzen.

Unternehmensziele erfassen

Die Unternehmensziele werden erfasst, entsprechend ihrer Realisierungshorizonte kategorisiert und innerhalb der Kategorien priorisiert. Außerdem sind inhaltliche (semantische) Beziehungen zwischen den Zielen festzulegen, sofern diese nicht durch den zeitlichen Ablauf und/oder die Priorisierung erkennbar sind.

Geschäftsprozesse erfassen

Die vom Projektanwender zu benennenden Geschäftsprozesse sind darzustellen und Abhängigkeiten sind zu erarbeiten und zu dokumentieren. Tatsächlich sind sich die Projektanwender selten ihrer Geschäftsprozesse und deren Abhängigkeiten bewusst. MAPKIT zielt nicht auf die Modellierung von Geschäftsprozessen, daher ist es an dieser Stelle ausreichend, die Geschäftsprozesse mit ihren temporären Abhängigkeiten zu erfassen, die unmittelbar mit dem zu untersuchenden IT-System zusammenhängen. Sollten sich durch die Geschäftsprozesse keine Vorgaben bzw. Randbedingungen für das IT-System ergeben, kann auf diese Teilphase ganz verzichtet werden.

Festlegung von Dienstgütevereinbarungen (SLAs)

Über ihre Festlegung und Dokumentation hinaus sind die Anforderungen der Geschäftsprozesse bzgl. Performance und Verfügbarkeit zu spezifizieren. Dabei ist darauf zu achten, dass die Anforderungen eindeutig und vor allem messtechnisch überprüfbar formuliert werden. Für alle kritischen Geschäftsprozesse sind die für einen reibungslosen Ablauf maximal erlaubten zeitlichen Beschränkungen festzulegen. Gegebenenfalls sind Anzahl und Umfang von Verletzungen dieser Schranken zu fixieren.

(b) Planung und Prognose

In dieser Phase werden Aktivitäten, Termine und Aufwände vereinbart, die zur Erstellung von Prognosen erforderlich sind. Dazu wird das Verhalten des existierenden Systems werkzeugunterstützt analysiert und bewertet, damit auf Basis der Ergebnisse das zukünftige Leistungsverhalten mit Hilfe von Prognosewerkzeugen bereits vor der Inbetriebnahme vorhergesagt werden kann.

Abbildung 03-4 stellt die Aktivitäten zur Planung und Prognose mit ihren Abhängigkeiten grafisch dar.

Zunächst wird eine Bestandsaufnahme des existierenden Systems vorgenommen, um ein möglichst exaktes Basis-Performance-Modell (kurz: Basismodell) der gegenwärtigen Situation entwickeln zu können. Es stellt den Ausgangspunkt für zukünftige Szenarien und deren Prognosemodelle dar. Es enthält Last- und Systembeschreibungen, die (teilweise) in das Prognosemodell übernommen werden können. Die Validation dieses Modells anhand des realen Systems schafft die notwendige Vertrauensbasis für die MAPKIT-Methode und die Prognose-Ergebnisse.

Als Randbedingungen für die Planung und Prognose sind folgende Situationen denkbar:

1. **Systemevolution**:
   Das existierende System wird in wesentlichen Teilen in ein zukünftiges System übernommen.

2. **Neuentwicklung**:
   Das zukünftige System unterscheidet sich so sehr vom existierenden, dass keine relevanten Erkenntnisse übernommen werden können. Im Extremfall handelt es sich um eine völlige Neuentwicklung.

Im ersten Fall werden aus der Ist-Situation die notwendigen Informationen für das Basismodell gewonnen. Im zweiten Fall muss in einem vorgeschalteten Schritt zunächst ein prototypisches System erstellt werden, das die Erfassung relevanter Informationen für das Basismodell ermöglicht.

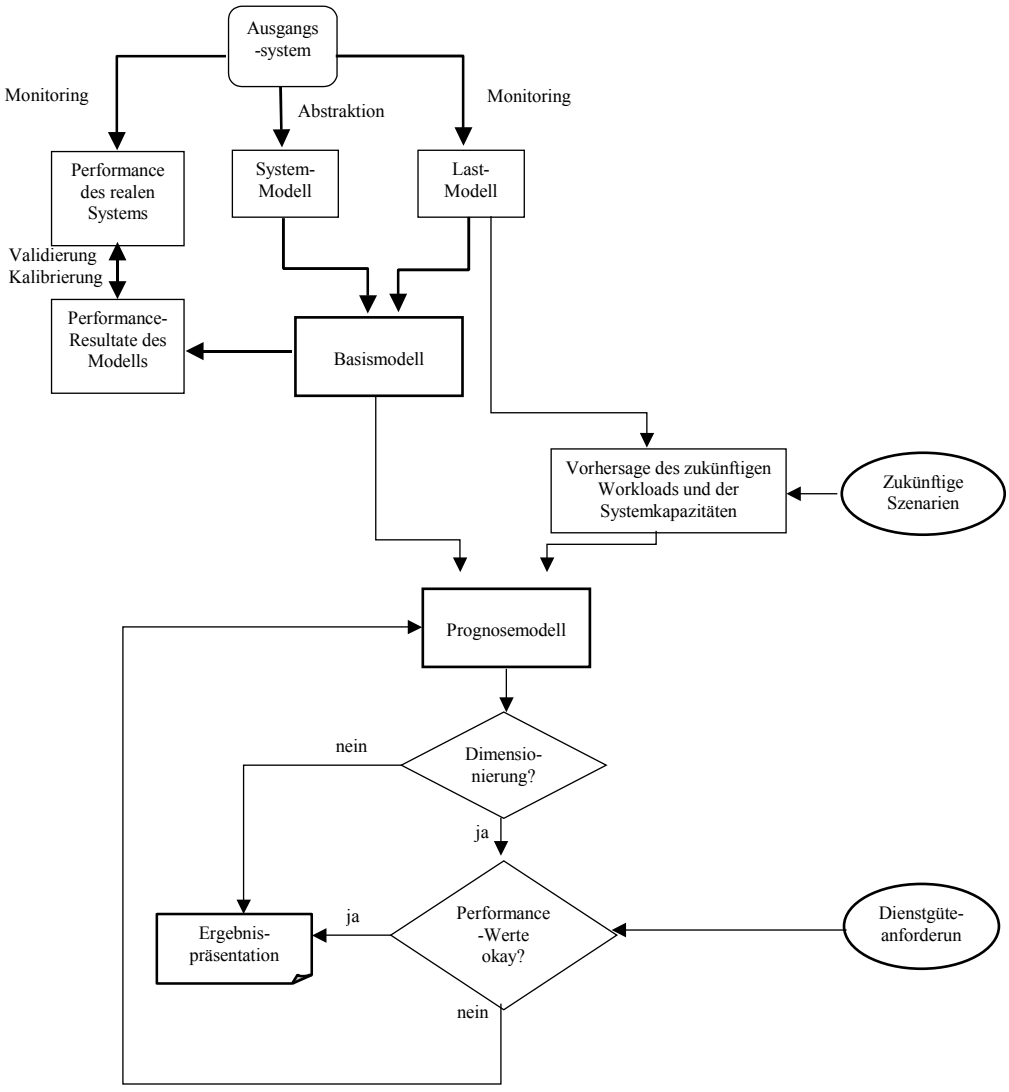


Abbildung 03-4: Aktivitäten zur Planung und Prognose

Nachfolgend werden die Teilaktivitäten der Phase „Planung und Prognose“ beschrieben.

Erstellung eines prototypischen Systems

Diese Aktivität ist nur dann durchzuführen, wenn eine Neuentwicklung des Systems geplant ist. In diesem Fall müssen die zukünftig beteiligten Systemkomponenten bereit gestellt und die IT-Prozesse (prototypisch) realisiert werden. Ein Beispiel für ein prototypisches System ist die Installation der zukünftigen Applikationen in einem vom Produktivsystem unabhängigen Testsystem.

Je genauer dieser Prototyp das zukünftige System abbildet, desto aussagekräftiger ist das im Anschluss zu entwickelnde Basismodell.

Erfassung von Komponenten

Zur Erstellung des Basismodells ist eine Bestandserfassung der Gegebenheiten beim Anwender vor Ort erforderlich. Diese Erhebung ist aus der Sicht des Kapazitätsmanagements durchzuführen, d.h. es sind nur performancerelevante Informationen zu erfassen. Es wird eine Übersicht über die verwendeten Komponenten eines IT-Systems erstellt. Dazu wird in einem ersten Schritt die Topologie und die sie tragenden Elemente wie Clients, Server und Netzinfrastruktur erfasst.

Clients werden mit ihrem Typ, dem verwendeten Betriebssystem, der Anzahl und ihren Standorten registriert. Darüber hinaus werden alle Applikationen benannt, die auf den Clients installiert sind. Die Server werden ebenfalls mit Typ, Betriebssystem, Mengen und Standorten registriert. Bei ihnen wird darüber hinaus festgehalten, für welche Anwendungen sie als Server fungieren.

Zur Erfassung der Netzinfrastruktur gehört die Dokumentation aktiver Elemente wie Switches, Router, Hubs, Gateways etc. sowie der verwendeten Protokolle. An dieser Stelle sind die Vernetzungsstrukturen und, falls vorhanden, logische Subnetze, virtuelle LANs etc. zu dokumentieren.

Nachfolgend sind die Kapazitäten der erfassten Komponenten zu ermitteln. Dazu werden für Netze die Bandbreiten, die verwendeten Paketgrößen und Protokoll-Overhead dokumentiert. Für die aktiven Elemente wie Router und Switches sind deren Vermittlungskapazitäten in Paketen/s oder alternativ die Bandbreiten interner Busse und gegebenenfalls die Speicherkapazitäten zu ermitteln. Die Leistungsfähigkeiten der vorhandenen Netzschnittstellen sind ebenfalls zu dokumentieren.

Für Clients und Server sind deren Kapazitäten applikationsspezifisch zu erfassen. Dazu zählen z.B. Angaben über Transaktionszahlen/s, NFS-Pakete/s, DB-Units/s, aber auch Angaben über die Leistungsfähigkeit von Netzkarten in Bytes/s. In jedem Fall ist die Anzahl vorhandener Prozessoren anzugeben. In Abhängigkeit vom Detaillierungsgrad der zu

erstellenden Modelle können für Clients und Server auch die Angaben über Hauptspeichergrößen, Plattenanzahl und -größen, Caches etc. von Interesse sein.

Alle hier erhobenen Angaben werden im sogenannten Systemmodell zusammengefasst.

Erfassung der IT-Prozesse

Die im System vorhandenen IT-Prozesse werden identifiziert, katalogisiert und gegebenenfalls grafisch dargestellt. Es ist ferner festzuhalten, welche IT-Prozesse durch welche Applikationen realisiert werden und um welchen Applikationstyp (z.B. Batch, Dialog) es sich handelt.

Abbildung der Geschäftsprozesse auf die IT-Prozesse

Der Zusammenhang zwischen den IT-Prozessen und den Geschäftsprozessen ist zu visualisieren. So sind Diagramme zu erstellen, die temporäre Zusammenhänge verdeutlichen. Die Performance-Anforderungen an die IT-Prozesse sind aus den Geschäftsprozessen mit Hilfe der Diagramme abzuleiten. Die Bewältigung dieser Teilphase erfordert einen nicht unwesentlichen manuellen Aufwand. Eine (semi)automatische Abwicklung ist kaum möglich.

Verkehrscharakterisierung und Lasterfassung

Die Kommunikationswege der Applikationen durch das vorhandene IT-System sind zu erfassen. Dabei sind die Komponenten des Systems und deren „Besuchsreihenfolgen“ zu dokumentieren. Einen weiteren wesentlichen Gesichtspunkt stellt die Ermittlung von Abhängigkeiten zwischen (Teil-) Applikationen dar. Damit sind Abhängigkeiten der folgenden Art gemeint: Wann immer die Transaktion A ausgeführt wird, ist 3-mal die Transaktion B auszuführen.

Weiterhin ist die Gesamtheit der User, die Verteilung auf die vorhandenen Lokationen sowie ein Tageslastprofil (Zugriffsmuster) der vorhandenen User zu erfassen. Die im System vorhandenen User sind in Benutzerklassen aufzuteilen. Die Aufteilung orientiert sich an den ausgeführten Applikationen bzw. an den ausgeführten Transaktionstypen. Für die Klassen sind typische Transaktions- bzw. Applikationsintensitäten zu bestimmen.

Darüber hinaus sind die Ressourcenverbräuche der Applikationen zu bestimmen. Dabei werden Transaktionsklassen bezüglich des benutzten Weges (Quelle-Ziel), des Ressourcenbedarfs an Komponenten und nach Prioritäten gebildet.

Als letztes sind solche Hintergrundlasten für zentrale Komponenten zu bestimmen, deren Herkunft im Detail nicht interessiert, deren Existenz aber einen signifikanten Einfluss auf die erreichbare Performance der kritischen IT-Prozesse und somit auf die Performance der

Geschäftsprozesse hat. Die Ausführung dieser Schritte führt zu einem detaillierten Lastmodell der im IT-System vorhandenen Applikationen.

Basismodell

Die Erstellung des Basismodells beinhaltet die Zusammenführung von Systemmodell und Lastmodell zu einem homogenen System (s. auch 03.01 (c) „Exkurs: Ansätze zur Systemabstraktion"). Die Erstellung eines Basismodells umfasst darüber hinaus die Validierung und Kalibrierung anhand erhobener Messdaten. Dazu zählen neben den user-orientierten Maßen wie Antwortzeiten und Durchsätze auch die systemorientierten Maße wie Auslastungen von oder Populationen an einzelnen Komponenten des Systems.

Zur Erstellung des Basismodells wird eine der in Abbildung 03-6 dargestellten Abstraktionsmethoden angewendet. Die Übergänge zwischen den einzelnen Klassen sind fließend.

Bei der Validierung bzw. Kalibrierung werden Performance-Ergebnisse des Basismodells mit den Messwerten des realen Systems verglichen. Falls sie hinreichend genau übereinstimmen ist das Basismodell validiert. Falls die Angaben zwischen Modell- und Messwerten nicht hinreichend genau übereinstimmen, kann eine Kalibrierung der Eingabe- oder Ausgabewerte stattfinden. So können zum Beispiel Hintergrundlasten, die noch nicht berücksichtigt wurden, zusätzlich eingebracht werden (Eingabekalibrierung).

An dieser Stelle ist eine Beurteilung des existierenden Systems und damit der Ausgangssituation vorzunehmen. Dazu gehören eine Engpassanalyse sowie die Überprüfung, ob definierte Dienstgüteanforderungen eingehalten werden. Zu dieser Beurteilung gehört aber auch die Analyse des Modellbildungsvorganges. Festgestellte Probleme bei der Erfassung von Ist-Daten oder der Erstellung von Modellen sollten dokumentiert werden, um diese zukünftig zu vermeiden.

Festlegung zukünftiger Lasten und Kapazitäten

Die Systemkapazitäten sind komponentenweise möglichst applikationsspezifisch zu prognostizieren. Dazu zählen z.B. Angaben über Transaktionszahlen/s, NFS-Pakete/s, DB-Units/s, aber auch Angaben über die Leistungsfähigkeit von Netzkarten in Bytes/s und die Anzahl vorhandener Prozessoren.

Die Vorhersage zukünftiger Lasten ist eine Schlüsselaktivität. Auf Basis der Lasten im laufenden Betrieb sind mit Hilfe von Abschätzungen und Trendberechnungen die zukünftig zu erwartenden Lasten zu prognostizieren. Weiterhin sind neue zu erwartende Anwendungen einzuplanen.

Festlegung der Dienstgüte

Für die zukünftigen Applikationen werden die einzuhaltenden Dienstgüteanforderungen und die dabei zugrunde gelegten Randbedingungen festgelegt. Zu den Randbedingungen gehören auch Zeitpläne, die angeben, wann die Systemmodifikationen vorgenommen werden sollen. Daraus wird ein Aktionsplan abgeleitet, in dem die mögliche Unterstützung durch ein Kapazitätsmanagement-Team festgehalten wird.

Es ist zu prüfen, ob die Dienstgüteanforderungen vollständig, realistisch und widerspruchsfrei sind. Vorhandene Lücken werden aufgezeigt und gegebenenfalls geschlossen. Es sollte eine Abstufung und/oder Priorisierung vorgenommen werden. Dabei ist zu untersuchen, welche der Vereinbarungen harte Schranken aufweisen und wo eventuell Sanktionen bei Nichteinhaltung vertraglich festgelegt sind.

Konzeptionierung von Prognosen

Das Prognoseparadigma (Stochastische Modellierung vs. Benchmarking) ist in Abhängigkeit des Basismodells zu wählen. Das Basismodell, welches die Ausgangssituation repräsentiert und validiert wurde, ist unter Berücksichtigung von Änderungen der Topologie und/oder der Last in ein Prognosemodell zu überführen. Weiterhin sind der Modellierungsgrad und der -umfang festzulegen.

Beim Experimentdesign von stochastischen Performance-Modellen ist zu berücksichtigen, dass vor allem numerische und simulative Modelle einen sehr hohen Rechenzeitbedarf für die Lösung besitzen. Hier sind vor allem die Anzahl der Modelläufe, also die Anzahl der Stützstellen im Parameterraum, zu minimieren. Als weitere Maßnahmen zur Komplexitätsreduktion bieten sich hierarchische Modellierung und Aggregierungstechniken an. Effiziente analytische Verfahren sind für Planungszwecke auf jeden Fall wesentlich günstiger. Falls möglich, sollten Sensitivitätsanalysen gemacht werden, um zu ermitteln, wie empfindlich ein System auf die Variation einzelner Eingangsparameter reagiert.

Bei einer Lastsimulation oder beim User-Benchmark ist zu entscheiden, ob alle vorgesehenen Komponenten und Systeme bereitgestellt werden müssen, oder ob aus Gründen der dafür anfallenden Kosten und hinsichtlich der Performance des Gesamtsystems auf weniger relevante Komponenten verzichtet werden kann. Wie bereits bei der Entwicklung des Basismodells ist zu überprüfen, ob der Einsatz virtueller Clients Verfälschungen der Performance-Werte verursachen kann.

Die Bildung von Benchmarks mit sehr unterschiedlichen Konfigurationen verursacht in der Regel deutlich höhere Kosten als die stochastische Performance-Modellierung. Bei der Auswahl der zu untersuchenden Szenarien muss daher besonders auf Aufwandseffizienz geachtet werden. Die Prognosen müssen so konzeptioniert werden, dass ihre Ergebnisse die

Überprüfung der vorher festgelegten Dienstgüte ermöglicht. Abbildung 03-5 stellt beispielhaft die Dienstgüteanforderung für die Antwortzeit verschiedenen Prognoseergebnissen gegenüber.

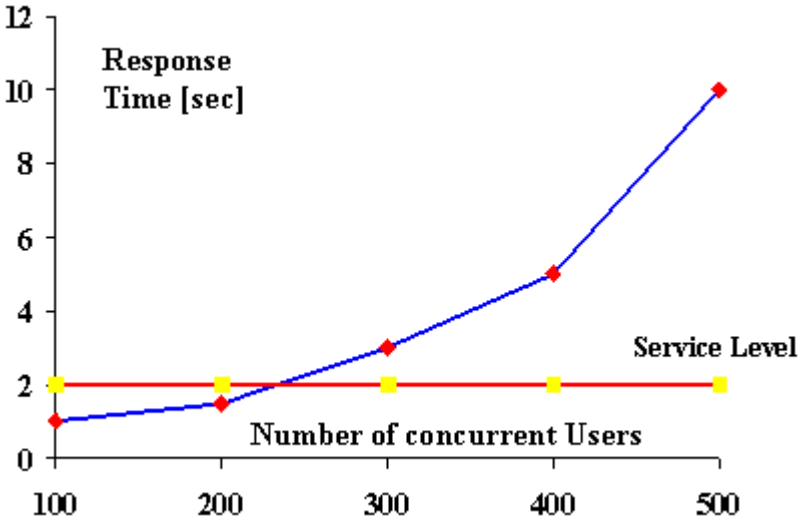


Abbildung 03-5: Antwortzeit und Dienstgüte

Experimente und Ergebnisbewertung

Der Planungsauftrag legt den Rahmen fest, innerhalb dessen die Modellexperimente zu planen sind und kann zwei unterschiedliche Zielrichtungen für die Experimente vorgeben:

Performance Prognose:

Die zu prognostizierenden Szenarien sind bzgl. Last und Systemkonfiguration relativ präzise vorgegeben. Für diese Szenarien sind Performance-Werte zu bestimmen. Die Prognoseergebnisse werden für die verschiedenen Szenarien direkt präsentiert. Wenn der Anwender mit den vorhergesagten Performance-Werten nicht zufrieden ist, sollte eine Dimensionierung angeschlossen werden.

Dimensionierung:

Die zu prognostizierenden Szenarien sind bzgl. der Last vorgegeben. Die Vorgaben für die Systemkonfiguration sind vage und legen lediglich den Rahmen fest, innerhalb dessen die den Anforderungen entsprechende Systemkonfiguration zu ermitteln ist. Die Prognoseer-

gebnisse werden mit den Dienstgüteanforderungen verglichen. Die Systemparameter werden (unter Berücksichtigung der vorgegebenen Rahmenbedingungen, insbesondere unter Berücksichtigung der verfügbaren Produkte und deren Kosten) so eingestellt, dass die gewünschte Dienstgüte erreicht wird. Das resultierende System wird anschließend als Empfehlung für die zukünftige Systemkonfiguration präsentiert. Sollten die Dienstgüteanforderungen nicht erfüllbar sein, ist zu prüfen, ob die Rahmenbedingungen oder die Anforderungen verändert werden müssen.

Präsentation der Ergebnisse

Es sind die Ausgangssituation, der Planungsauftrag und die entsprechenden Ergebnisse darzustellen. Dabei kann es sich um eine Performance-Übersicht des aktuellen und zukünftigen Systems oder um eine Dimensionierungsempfehlung handeln. Es sind alle Voraussetzungen, die den Experimenten zugrunde liegen, kurz darzustellen und deren Auswirkungen auf die erzielten Ergebnisse zu erläutern.

Erzielte Ergebnisse oder gewonnene Erkenntnisse, die nicht zum unmittelbaren Planungsauftrag oder zu dem Analysefokus des aktuellen Systems gehören, aber kritische oder sogar katastrophale Auswirkungen auf das System, bzw. Teile des Systems haben, müssen dem Anwender aus Fürsorgepflicht mitgeteilt werden.

Die Ergebnisse sind für den jeweiligen Adressatenkreis zu verdichten bzw. zusammenzufassen. So interessieren sich die Netzwerkadministratoren für die Auslastungen ihrer Komponenten und Durchlaufzeiten von Paketen durch Netze, während z.B. die Anwender bzw. deren Management an Antwortzeiten ihrer Applikationen interessiert sind. Das Firmenmanagement hingegen betrachtet eher die Einhaltung der Performance-Anforderungen auf Geschäftsprozess-Ebene.

Falls mehrere Systemalternativen vorliegen, so sind Kriterien anzubieten, die die Anzahl der Alternativen weiter reduzieren.

(c) Exkurs: Ansätze zur Systemabstraktion

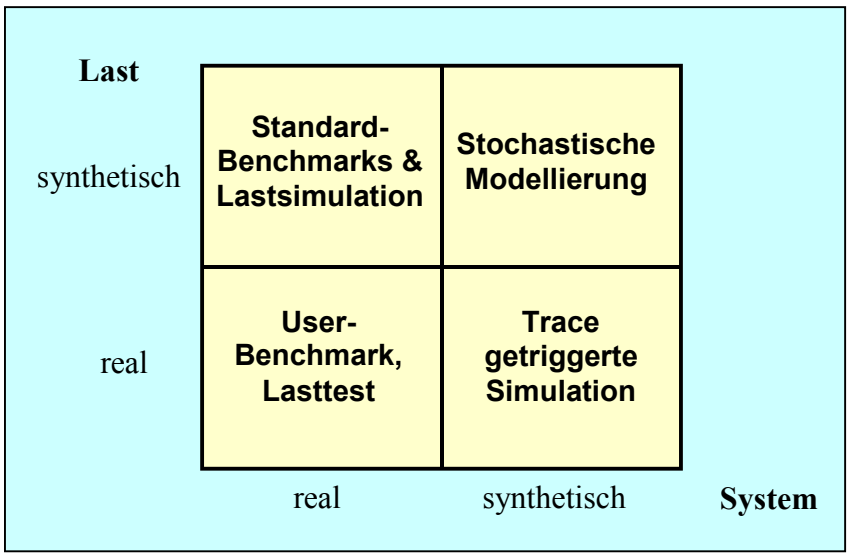


Abbildung 03-6: Klassifizierung von Systemabstraktionen

Stochastische Performance Modellierung

Als erstes sind der Modellierungsgrad und der Modellierungsumfang festzulegen, d.h. es ist zu entscheiden, welche Aspekte des realen Systems im Modell abgebildet werden sollen und auf welcher Ebene (Anwendung, Netzwerk etc.) modelliert wird. Eng damit verbunden ist die Festlegung der Lösungstechnik (Modellklasse) und letztlich eines Modellierungs-Tools.

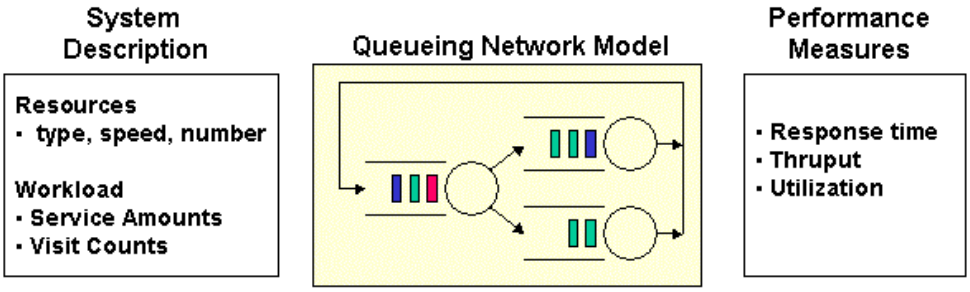


Abbildung 03-7: Der Modellbildungsprozess am Beispiel von Warteschlangensystemen

Im Falle simulativer Lösungstechniken sind geeignete Performance-Maße festzulegen, bei denen die Erreichung vordefinierter Konfidenzlevel den Abbruch der Simulation bedeutet. Weiterhin ist bei stationären Untersuchungen die Länge der Einschwingphase (warm-up) zu berücksichtigen. Analog gilt dieses für die Definition von Genauigkeitswerten bei approximativen oder numerischen Lösungstechniken.

Lastsimulation und Kunden-Benchmarking

Neben der Modellierung gewinnen in der letzten Zeit die Verfahren der Lastsimulation und das Kunden-Benchmarking immer mehr an Bedeutung.

Lastsimulation:

Lastsimulation beinhaltet die Nachbildung realer Applikationen und Datenbestände sowie des Benutzerverhaltens. Im Realbetrieb werden die Lastdaten erhoben, mit denen die Clients die Netzwerke und die Server-Systeme belasten. Darüber hinaus werden die Lastdaten erhoben, mit denen die Serversysteme die Netzwerke belasten. Die Synthetisierung erfolgt mittels Lastgeneratoren, die auf Basis dieser Lastdaten die entsprechenden Netz- und Serverlasten produzieren. Der Ablauf erfolgt i.d.R. in einer der Realität nachgebildeten Testumgebung. Mit Messwerkzeugen wird das Systemverhalten hinsichtlich Verfügbarkeit, Zuverlässigkeit und Performance analysiert, wodurch die Bewertung und Dimensionierung des Gesamtsystems ermöglicht wird. Der Aufwand ist vor allem für die Auswertung der erhobenen Last- und Leistungsdaten und deren Umsetzung in die synthetischen Lastprofile sowie für die Validierung und Kalibrierung dieser Profile sehr hoch. Deshalb wird die Lastsimulation insbesondere zur Analyse des Systemverhaltens von vernetzten IT-Strukturen eingesetzt. Sie wird eingesetzt, wenn sensible Daten vorliegen oder wenn die realen Applikationen nicht eingesetzt werden können. Sie wird außerdem zur Generierung von Hoch- und Extremlastfällen eingesetzt, um nicht nur die Performance, sondern auch die Verfügbarkeit und Zuverlässigkeit der System- und Netzwerkkomponenten zu analysieren und zu bewerten.

Kunden-Benchmarking:

Benchmarks umfassen eine Menge von “repräsentativen“ Programmen oder Anwendungen und werden unterschieden nach “Standard-Benchmarks“ und “Kunden-Benchmarks“.

Standard-Benchmarks sind offiziell standardisiert (u.a. SPEC, TPC) und dienen insbesondere zum Performance-Vergleich unterschiedlicher IT-Systeme. Beim Upgrading werden die Benchmark-Ergebnisse häufig zur Prognose der Skalierbarkeit und zur Dimensionierung herangezogen. Ihre Aussagekraft liegt zwischen denen von Daumenregeln und Trendanalysen.

Kunden-Benchmarks bauen auf realen Applikationen und Datenbeständen auf und stellen i.d.R. ein repräsentatives Abbild der realen Last dar. Sie werden sehr häufig zur Absicherung einer Performance-gerechten Dimensionierung von Server-Systemen verwendet. Zur Nachbildung der realen Benutzer und deren Verhalten werden häufig Treiber-Systeme eingesetzt, die in die Lage sind, auf Basis von vorher am Client-PC aufgezeichneten, repräsentativen Benutzerdialogen, eine Vielzahl von Benutzern synthetisch nachzubilden. Die so erzeugte Last lässt sich vervielfältigen. Abhängig von der real darunter befindlichen Systemumgebung ergibt sich das an der Realität nachgebildete Lastverhalten (vgl. Abbildung 03-8).

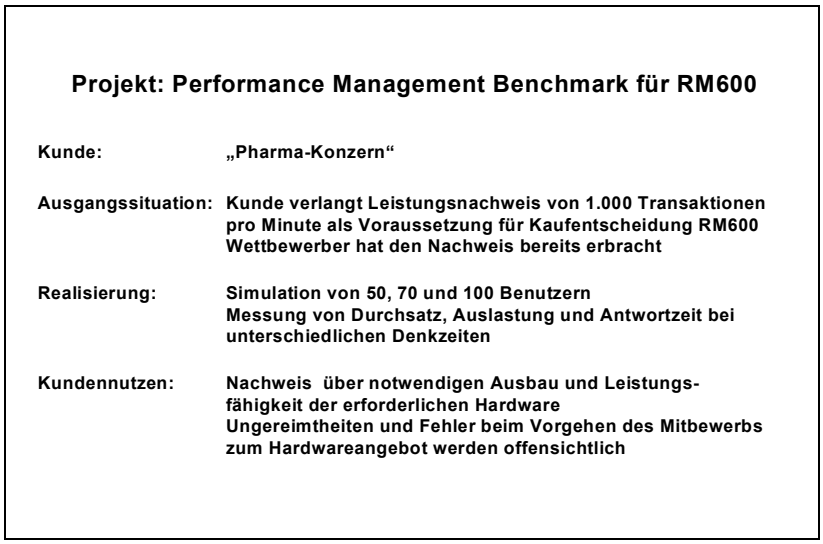

Projekt: Performance Management Benchmark für RM600
Kunde:
„Pharma-Konzern“
Ausgangssituation:
Kunde verlangt Leistungsnachweis von 1.000 Transaktionen pro Minute als Voraussetzung für Kaufentscheidung RM600
Wettbewerber hat den Nachweis bereits erbracht
Realisierung:
Simulation von 50, 70 und 100 Benutzern
Messung von Durchsatz, Auslastung und Antwortzeit bei unterschiedlichen Denkzeiten
Kundennutzen:
Nachweis über notwendigen Ausbau und Leistungsfähigkeit der erforderlichen Hardware
Ungereimtheiten und Fehler beim Vorgehen des Mitbewerbs zum Hardwareangebot werden offensichtlich

Abbildung 03-8: Kunden-Benchmark

Zur Dimensionierung des Netzwerkes und damit des Gesamtsystems ist u.a. wegen der Sequentialisierungs-Effekte der Einsatz eines Treibersystems nur bedingt geeignet. Last- und Performance-Tests in vernetzten IT-Umgebungen werden deshalb zweckdienlich mit einer großen Anzahl (>> 100) von Clientsystemen durchgeführt, um so möglichst realitätsnah und vielseitig Lasten erzeugen zu können. Insbesondere ist die Einbeziehung des Netzwerkes bei solchen Tests ein nicht zu vernachlässigender Aspekt, dem nur mit der Beteiligung von sehr vielen Netzknoten (Clients) Rechenschaft geleistet wird. Erst die Möglichkeit, an vielen Clients Lasten zu erzeugen, die alle nahezu gleichzeitig oder auch nur sporadisch auf das Netzwerk und die Server-Systeme zugreifen, können Funktionalität

und Performance des vorhandenen Netzes und somit das Verhalten des Gesamtsystems entsprechend bewertet werden.

Die Durchführung derartiger Last- und Performance-Tests macht es erforderlich, dass Mechanismen zur definierten Lastgenerierung sowie zur Ansteuerung und Überwachung der PCs verfügbar sind. Diesbezüglich bestehen spezifische Anforderungen an die Testautomatisierung hinsichtlich Lasterzeugung, Reproduzierbarkeit und Variierbarkeit sowie Konsistenz und Überprüfbarkeit der Ergebnisse. Bei Beteiligung vieler Clients empfiehlt sich eine Messautomatisierung in Form einer zentralen Steuerungseinheit zur Synchronisation der Clients und Steuerung der gesamten Umgebung, um die Kontinuität in der Erzeugung einer gleichförmigen Last zu gewährleisten.

Last- und Performance-Tests

In modernen IT-Strukturen bestehen starke Interdependenzen zwischen den Aspekten der Verfügbarkeit, der Zuverlässigkeit und der Performance. So kann es insbesondere in Hochlastfällen zum Reorganisieren von Netzwerkkomponenten und zum Rebooten von Server-Systemen sowie sogar zum Systemstillstand kommen. Deshalb werden insbesondere in vernetzten heterogenen Strukturen vor der Inbetriebnahme des Realbetriebs Last- und Stresstests durchgeführt, um das Systemverhalten zu analysieren und zu bewerten.

Diese Tests erstrecken sich i.d.R. auf die Einbeziehung der Netzwerke und deren Komponenten sowie einer großen Anzahl Clients. Mit der Durchführung von Lasttests wird für den Anwender ein Mehrwert generiert, der sich wie folgt darstellt:

@ Bedarfsgerechte Dimensionierung der Server-Systeme (Rightsizing) vor dem Realbetrieb (auch für unterschiedliche Konfigurationen)

@ Aussagen im Wesentlichen zum CPU-, Platten- und Hauptspeicherbedarf jeweils für verschiedene Lastfälle (applikations-spezifisch)

- nach längerer Aktivzeit
- mit der Möglichkeit der Ermittlung des Ressourcenbedarfs bei unterschiedlichen Lastzusammenstellungen

@ Analyse und Eliminierung von Engpässen vor der Aufnahme des Produktivbetriebs (keine Störungen im Realbetrieb)

@ Ermittlung des Systemverhaltens (Verfügbarkeit, Zuverlässigkeit, Performance) insbesondere in Hochlastsituationen vor Aufnahme des Realbetriebs („Stresstest“). Daraus resultiert nicht nur eine Kostenverlagerung, sondern auch eine Kostenreduzierung, da so die bedarfsgerechte Dimensionierung und die Aufnahme des störungsfreien Produktivbetriebs ermöglicht werden (vgl. Abbildung 03-9).

Projekt: Last- und Performancetest IT2000

Kunde:	**„Bundesamt“**
Ausgangssituation:	**Kunde verlangt Leistungsnachweis von:** **- Können 4 500 Transaktionen pro Stunde verarbeitet werden?** **- Kann Reliant UNIX die aufgerufenen Prozesse bearbeiten?** **- Über welche CPU-Power muss der DB-Server verfügen?**
Realisierung:	**- Simulation von 1 500 Benutzern auf 240 physikalischen PCs** **- Ermittlung von Durchsatzraten, CPU- und DB-Last mittels einer vollautomatisierten Test-, Mess- und Auswerteumgebung** **- Überprüfung des Antwortzeitverhaltens**
Kundennutzen:	**- Funktionaler Test des Gesamtsystems** **- Aufdecken von Designproblemen** **- Nachweis über geforderten Durchsatz und Antwortzeiten** **- Dimensionierung des Gesamtsystems** **- Deutliche Performance-Verbesserungen durch DB-Optimierung**

Abbildung 03-9: Last- und Performance-Test

(d) Überwachung

Sowohl Service User (Endbenutzer) als auch Service Provider (Dienstleister) sind interessiert an der Ausführung der Geschäftsprozesse unter akzeptablen Randbedingungen, d.h. an Spezifizierung und Überprüfung der Dienstgüte (Quality of Service) hinsichtlich

- Verfügbarkeit,
- Zuverlässigkeit und
- Performance (Antwortzeit, Durchsatz).

Insgesamt soll sich eine ausgewogene Nutzung der Systemressourcen unter Einhaltung der vereinbarten Dienstgüte ergeben. Das beinhaltet die ständige Überwachung des laufenden Betriebs unter technischen und wirtschaftlichen Gesichtspunkten. Dazu ist es erforderlich, das Verhalten der beteiligten Systeme und ihrer Applikationen durch ein umfassendes Monitoring zu überwachen. Überwacht werden das Gesamtsystem hinsichtlich der vereinbarten Dienstgüte sowie die Systemkomponenten hinsichtlich

- Auslastungsgraden,
- Verweilzeiten,
- Warteschlangenlängen und
- Funktionsaufrufe pro Zeiteinheit.

So können Trends abgeleitet und mögliche Engpässe rechtzeitig erkannt werden.

Überwachungsobjekte

Dienstgüte (Quality of Service)

Zur Ermittlung und Überwachung der Dienstgüte wird häufig die in Frameworks verwendete Agententechnologie eingesetzt. Kollektoren müssen auf den Clients installiert werden und instrumentieren die zu vermessende Applikation an den für die Antwortzeit relevanten Messstellen. So können die Antwortzeiten ermittelt und anhand vorgegebener Schwellwerte für Warnungen und Alarme überwacht werden. Beim Überschreiten dieser Schwellwerte werden diese an die zentrale Konsole gemeldet und gemäß Ampelsemantik (grün für befriedigende Dienstgüte, gelb für Warnung und rot für Alarm) sowie mit Ort und Art der Verletzung angezeigt.

Eine andere Art der Ermittlung und Überwachung der Dienstgüte basiert auf der Anwendung von Referenz-PCs. Der Referenz-PC beinhaltet einen SW-Treiber („Testautomat“), der einen realen Benutzer an der grafischen Oberfläche simuliert. Business-kritische Transaktionen werden in festzulegenden Abständen durchgeführt und deren Antwortzeit gemäß DIN 66273 ermittelt und protokolliert. Für jede Transaktion kann ein Schwellwert (= maximal tolerierbare Dauer der TA; siehe Dienstgütevereinbarung bzw. Service Level Agreement (SLA) zwischen IT-Betreiber und Fachabteilung) definiert werden. Bei Schwellwertüberschreitungen am Client wird neben einem Alert eine entsprechende Meldung an den Administrator gesendet. Vorteil gegenüber der Agententechnolgie ist u.a. die objektiv messbare Antwortzeit direkt an der Mensch-Maschine-Schnittstelle.

Ressourcennutzung

Bei Erkennung von Dienstgüteverletzungen muss eine globale Analyse hinsichtlich vorhandener Engpässe und Schwachstellen durchgeführt werden. Es muss eine permanente Überwachung des Lastaufkommens und der Auslastung aller vorhandenen Komponenten stattfinden. Dazu ist ein integrierter Messansatz zur automatischen Überwachung aller relevanten Systemkomponenten erforderlich. Dies wird i.d.R. durch die Resource and Performance Management Systems in den Frameworks bewerkstelligt. Dort werden einerseits basierend auf der Agententechnologie auf den Server- (und evtl. auch auf den Client-) Systemen mittels Kollektoren Performance-Daten in den Betriebsystemen, Datenbanken

und auch Applikationen (z.B. R/3) erhoben und abgespeichert, die dann ausgewertet und grafisch aufbereitet werden. Andererseits wird im Netzwerkbereich basierend auf der RMON-Technik ein Remote-Monitoring per Hardware-Messstellen in den aktiven Netzkomponenten durchgeführt. Alle gesammelten Informationen müssen zentral zusammengeführt und sollten dort korreliert ausgewertet werden.

Der Einsatz von Frameworks bringt auch spezielle Nachteile mit sich, z.B. sind Frameworks nicht immer verfügbar sowie sehr komplex und kostenintensiv. Es fehlt die individuelle Erweiterbarkeit der Mess- und Analysewerkzeuge, und die Instrumentierbarkeit der Software ist nicht immer gegeben. Dies führte im Rahmen von MAPKIT zur Entwicklung der automatisierten Mess- und Auswerteumgebung PERMEV, die nicht nur zur Überwachung, sondern auch zur Detail- und Ursachenanalyse eingesetzt wird. Auf sie wird im folgenden Kapitel näher eingegangen.

(e) Analyse und Tuning

Die Performance eines IT-Systems ist als die Fähigkeit definiert, gegebene Anforderungen an Antwortzeiten und Datendurchsatz zu erfüllen. Hinter dem Begriff Performanceoptimierung verbirgt sich ein Prozess, der im Wesentlichen die folgenden drei Schritte umfasst:

- Systematische Identifizierung und Analyse von Problemen
- Erarbeitung und Festlegung von Tuningmaßnahmen
- Realisierung der vorgeschlagenen Maßnahmen und Kontrolle deren Auswirkung

Ziel des Tunings ist eine Feinabstimmung des Systems, um eine optimale Nutzung der Systemressourcen unter Einhaltung einer vorgegebenen Servicegüte sicherzustellen.

Vom praktischen Ansatz her lassen sich Tuningmaßnahmen in folgende Kategorien einteilen:

- Technisches Tuning
- Gegenstand des technischen Tunings sind
  - Hardware: Server und Clients, Netzwerk und Datenkommunikationssysteme,
  - Betriebssysteme für Server und Netzwerk sowie systemnahe Software, wie Datenbanken

  Durch das technische Tuning werden alle zum Gesamtsystem gehörenden Komponenten so eingestellt, dass die anfallende Last vom System optimal verarbeitet werden kann und sich keine Performance-Engpässe bilden.

@ Applikationstuning

- auf Programmebene:
  Der Schwerpunkt ist die Überprüfung anwendungsspezifischer Vorgänge mit dem Ziel, den Ressourcenbedarf an CPU, Plattenzugriffen und Netzwerktransfer zu reduzieren.

- durch organisatorische Eingriffe:
  Gemeint ist hier eine Optimierung des IT-Betriebs in Bezug auf ausgesuchte Geschäftsprozesse. So können Kapazitätsengpässe durch veränderte zeitliche Staffelung im Betrieb mit einer Verlagerung von Abläufen in betriebsarme Zeiten entschärft werden. Des Weiteren ist es mitunter sinnvoll, ressourcenintensive Prozesse nur in begrenzter Parallelität oder für privilegierte Benutzer zuzulassen.

Nach der Festlegung des Untersuchungszieles verschafft sich ein Performance-Experte bei einer Leistungsuntersuchung im Rahmen einer Globalanalyse einen Überblick zur Situation im Gesamtsystem. Ergeben sich hieraus keine Indizien für Engpässe, so sind im nächsten Schritt einer Detailanalyse top-down die problembehafteten Komponenten oder Applikationen zu identifizieren.

Zu diesem Zweck sind während repräsentativer Zeitintervalle Messungen durchzuführen und diverse Systemparameter zu überprüfen. Wichtig hierbei ist, Zahlen herauszufiltern, die folgendes beschreiben:

@ das gesamte Systemverhalten

@ den Zustand einzelner Systemkomponenten

@ das Verhalten einzelner Applikationen

@ den zeitlichen Trend

Je nach Aufgabenstellung sind im Detail unterschiedliche Kenngrößen zu ermitteln, wie:

@ Auslastung von Prozessoren und anderen Geräten

@ Leistungsfähigkeit dieser Geräte oder Medien

@ Nutzung von verfügbaren Speichereinheiten (Memory, Cache, Disk)

@ Häufigkeit und Dauer von Datei-, Platten- und Netzzugriffen mit zugehörigen Datenvolumina

- Häufigkeit von Systemaufrufen

- Task-/Prozessauslastung und Prozessstruktur
- Ressourcenbedarf von Geschäftsprozessen auf unterschiedlichen HW-Komponenten
- Antwortzeiten und Ankunftsraten von Geschäftsprozessen
- Wartezeiten und Sperrverhalten
- Kommunikationsverhalten und Synchronisationsmuster bei verteilter Verarbeitung
- Umschalt- und Latenzzeiten

Obige Aufstellung gibt einen globalen Überblick zu wichtigen Kenngrößen und stellt keinen Anspruch auf Vollständigkeit. Es wird allerdings deutlich, dass bei der Durchführung der angesprochenen Top-down-Analyse zur richtigen Interpretation des verfügbaren Datenmaterials viel Erfahrung und profundes Know-how erforderlich ist.

Neben der Ursachenanalyse ist es wichtig, die Auswirkung von umgesetzten Tuningmaßnahmen zu quantifizieren. Es empfiehlt sich hierbei, nur Schritt für Schritt vorzugehen, um so jeweils den Einfluss einzelner Maßnahmen zu überprüfen. Durch eine problemadäquate Toolunterstützung lassen sich die Zeitaufwände zur Ursachenanalyse in vielen Fällen deutlich reduzieren. Neben einer komfortablen Messdatenerfassung ist eine zielgerichtete Auswertung, Filterung, Verdichtung und Darstellung von Informationen hilfreich, um den Blick für das Wesentliche freizuhalten und die richtige Interpretation von komplexen Zusammenhängen zu erleichtern.

Zur Vertiefung geht das nächste Kapitel ein auf Methoden und Ansätze bei der Messdatenerhebung und stellt eine Reihe hilfreicher Tools und Produkte vor, die das Vorgehen bei Analyse und Tuning wesentlich erleichtern.

(f) Beispiel R/3-Kapazitätsmanagement

Aus dem Umfeld des R/3-Kapazitätsmanagements kennen wir das in Abbildung 03-10 gezeigte Szenario.

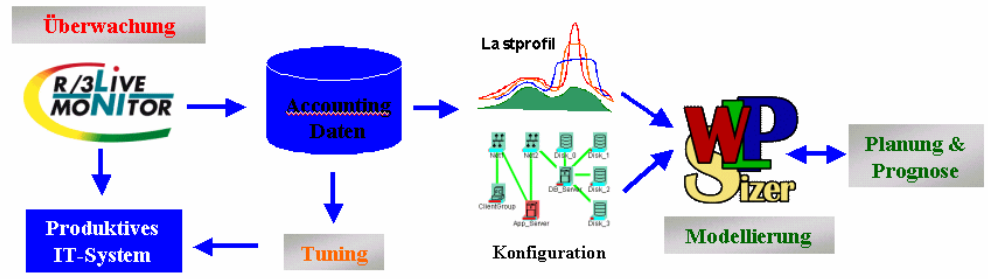


Abbildung 03-10: R/3-Kapazitätsmanagement mit Live-Monitor und WLPSizer

Aus der Systemüberwachung des laufenden SAP R/3 Produktivbetriebs oder eines Stresstests werden zum einen die im R/3 System protokollierten Statistik- bzw. Accountingsätze mit dem R/3 Live Monitor erfasst und zum anderen Betriebssystem- und Datenbankstatistiken gesammelt.

Diese Daten sind die Grundlage für die nun folgende Performance- und Engpassanalyse, aus der sich in der Regel Tuningempfehlungen ableiten lassen. Aus der Klassifizierung der Statistiksätze ergibt sich die Lastbeschreibung in Form von sogenannten Lastprofilen. Erst wenn die Tuningempfehlungen umgesetzt sind und keine gravierenden Engpässe mehr vorhanden sind, wird nach einer erneuten Messung aus den Lastprofilen und den Informationen über die Hardware- und Softwarekonfiguration, in Abbildung 03-10 kurz mit Konfiguration bezeichnet, ein Basismodell erstellt und kalibriert. Dies geschieht mit dem Modellierungs- und Analysewerkzeug WLPSizer.

Durch Modifikation der Lastprofile im Sinne einer Lastprognose können Prognosemodelle erstellt und mit WLPSizer gelöst werden.

03.02 Verfahren

(a) Generelle Aspekte

Verfahren müssen einerseits die Überwachung und Analyse der Ist-Situation und andererseits die Prognose zukünftiger Szenarien aus Gesamtsystemsicht unterstützen. Diese generelle Forderung hat vielfältige Konsequenzen.

Eigenentwicklung anstelle von Werkzeugeinkauf

Auf Grund der Erfahrungen im MAPKIT-Projekt hat sich die Notwendigkeit gezeigt, eigene Modellierungs- und auch Mess- und Analysewerkzeuge zur Kapazitätsplanung zu entwickeln bzw. weiterzuentwickeln. Die Lizenz- und Preispolitik der meist US-

wickeln bzw. weiterzuentwickeln. Die Lizenz- und Preispolitik der meist US-amerikanischen Hersteller und Distributoren erschwert die Umsetzung eines erfolgreichen Kapazitätsmanagements in der Praxis ganz erheblich. Dadurch gewinnen Eigenentwicklungen und der Einsatz von individuell anpassbaren elementaren und frei verfügbaren Werkzeugen an Bedeutung. Beispiele sind die Modellierungswerkzeuge VITO, WLPSizer und das Monitoring Tool PERMEX/PERMEV.

Konsistente Globalsicht

Plattformübergreifende Sicht

Ein umfassendes Kapazitätsmanagement beinhaltet die Analyse und Bewertung des Gesamtsystemverhaltens aus Endbenutzersicht. Zur Durchführung von Messungen ist eine globale Zeitbasis erstrebenswert, was allerdings in (weiträumig) vernetzten, heterogenen IT-Systemen schnell an Grenzen stößt. Für die weiterverarbeitenden Analyse- und Modellierungswerkzeuge müssen die Messwerkzeuge die Messdaten in einem plattformneutralen Format liefern.

Zeitliche Zuordnung

Für die Verfolgung von mehrstufigen Vorgängen ist eine eindeutige zeitliche Zuordnung der einzelnen Teilschritte notwendig (globale systemweite Zeitbasis). Dazu sind entsprechende Uhrensynchronisationen zu entwickeln und einzusetzen. Erst damit ist es möglich, Vorgangsdauern aus Teilschritten, die auf unterschiedlichen Plattformen und in Netzen ablaufen, zu einer Gesamtantwortzeit zusammenzusetzen.

Unterschiedliche Messebenen (u.a. Netz: ISO-Ebenen; Gesamtsystem: Applikationen)

Wünschenswert ist die Analyse des Gesamtsystems aus Endbenutzersicht. Dazu sind Messungen und Analysen des Systemverhaltens auf unterschiedlichen Systemebenen erforderlich. Im Netzwerk werden häufig Analysen auf den unteren ISO-Ebenen aber auch auf Ebene 7 (z.B. SMB-Statistiken) durchgeführt. Bei den Server-Systemen und vor allem im Gesamtsystem sind Messungen auf unterschiedlichen physikalischen Levels und auch auf logischer (Geschäftsprozess-) Ebene von Interesse.

Zuordnung von logischen Elementen zu physikalischen Ressourcen

Ziel ist es, die Zuordnung von Geschäftsprozessen und den IT-Prozessen bzw. Transaktionen, die zu deren Abarbeitung erforderlich sind, zu den physikalischen Ressourcen-Verbräuchen herzustellen. Dazu sind erforderlich:

Transaktionsverfolgung und -analyse

Die Messungen müssen transaktionsorientiert erfolgen. Eine Messung von Ressourcen-Verbräuchen unterhalb der Ebene von Transaktionen (Datenpakete, Prozesse, I/O-Operationen) ist nicht ausreichend. Ansätze dazu basieren auf der Agententechnologie und ähnlichen Ansätzen (u.a. Optimal Application Expert).

Verursachergerechte Zuordnung von Ressourcen-Verbräuchen

Messungen müssen applikationsorientiert erfolgen, d.h. Ressourcen-Verbräuche müssen den Applikationen und Transaktionen zugeordnet werden können. Für die Kapazitätsplanung ist z.B. eine pauschale Aussage über die gesamte Bandbreitennutzung in den Netzen (LAN-Segment, Backbone, WAN) nicht ausreichend, wird aber wie auch die Komponentenauslastung als Hilfsmittel herangezogen.

Häufig werden die Informationen aus Accounting-Systemen genutzt. Diese Systeme sind spezielle, in das Betriebssystem und auch in Applikationspakete integrierte Software-Monitore. Ihre Aufgabe ist die Erfassung und Abrechnung der von Programmen und/oder Benutzeraufträgen benötigten Betriebsmittel. Bei den sog. "offenen" Betriebssystemen (UNIX, Windows-NT) ist allerdings eine Zuordnung von prozessspezifischen Ressourcenverbräuchen zu verursachenden Transaktionen und damit zu Geschäftsprozessen wegen der Mehrfachnutzung eines Prozesses durch mehrere Transaktionen kaum möglich. Dagegen sind die Accounting-Systeme in Applikationspaketen (z.B. SAP R/3) besser geeignet. Hier ist auch oft die Erfassung benutzerspezifischer oder systeminterner Informationssätze möglich. So lassen sich auch system- und anwendungsbezogene Last- und Leistungsdaten ableiten.

Analyse von Dienstabwicklungsdauern für Clients, Server und Netze

Verweilzeiten an den Servern (Service- und Wartezeiten) und Netzlaufzeiten bilden wichtige Basisinformationen für das Kapazitätsmanagement. Wichtige Beispiele sind die Ermittlung der Vorgangsdauer bei der Software-Verteilung und die Bestimmung von Antwortzeiten im OLTP-Betrieb sowie insbesondere in vernetzten heterogenen Umgebungen. Dafür gibt es teilweise verheißungsvolle Ansätze (z.B. Optimal Application Expert), die aber in High Speed- und geswitchten Netzen nicht mehr greifen.

Instrumentierbare Software (Software-Sensoren)

In vernetzten heterogenen Systemen gibt es nicht mehr den von der proprietären Welt bekannten klassischen "Transaktionsbegriff". Damit ist auch die oben aufgeführte Zuordnung äußerst schwierig. Ein möglicher Lösungsweg ist neben der Nutzung von prozessspezifischen Messdaten (z.B. unter UNIX ps und acct) die Instrumentierung von Applikations-Software. Ansätze dazu werden u.a. mit ARM (Application Response Measurement) ver-

folgt, wo ein Standard API für messbare Software spezifiziert wird. Eine andere Möglichkeit wird mittels Hybrid-Monitoring (kombiniertes Hardware- und Software-Monitoring) angegangen.

Statistische Unabhängigkeit und Skalierbarkeit

Sampling (Stichprobenverfahren)

Dieses Verfahren basiert auf Serien von Aufzeichnungen der Systemzustände zu vom Objektgeschehen unabhängigen Zeitpunkten. Dabei müssen die Systembedingungen und die Last relativ konstant sein, und es darf keine Abhängigkeit zwischen Stichprobe und Ereignis bestehen.

Vorteil: Viele Analysen benötigen Messdaten, die über längere Zeiträume mit geringem Mess-Overhead erhoben werden.

Tracing (Event Driven Monitoring)

Die kontinuierliche Aufzeichnung von ausgewählten Ereignissen mit Angabe von Ort, Zeitpunkt, Spezifikation des Ereignisses liefert folgende Einsicht: Was geschieht wo, wann und warum.

Vorteil: Umfangreich (aber: hoher Mess-Overhead), detailliert, Zuordnung von Ort, Zeitpunkt, Ereignisauslöser und Auswirkung. Je nach Zielsetzung und Aufgabenstellung müssen die unterschiedlichen Monitore sowohl im Sampling Mode als auch im Tracing Mode arbeiten.

Mess-Overhead (Skalierbarkeit)

Maßgeblich hierfür ist die Zielsetzung der Messung: Für die Überwachung und Globalanalyse ist es wichtig, dass der Mess-Overhead und vor allem die Menge der erhobenen Messdaten nicht zu groß werden, d.h. relativ große Messintervalle und nur wenige Messindizes. Dagegen verlangen Ursachenanalysen und die Ermittlung von Last- und Performance-Daten für die Prognose gezielte und detaillierte Messungen, das beinhaltet i.d.R. die Erhebung von vielen und exakten Messwerten über relativ kleine Zeiträume.

Automatisierung versus individuelle Steuerung

Steuerung von Messwerkzeugen

Abhängig von Zielsetzung und Zielgruppe sind folgende Steuerungen denkbar:

- Weitestgehend automatisierte Überwachung und Globalanalyse
- Detail-/Ursachenanalyse und Ermittlung des transaktionsspezifischen Ressourcenbedarfs ist sowohl automatisiert als auch individuell steuerbar

Problem**:** Verschiedene Software-Monitore werden z.B. unter UNIX zeitversetzt gestartet, laufen unsynchronisiert ab und liefern unterschiedliche, unkorrelierte Messdaten. Eine umfassende Systemanalyse ist so ohne zusätzliche Synchronisierungsmaßnahmen nahezu unmöglich.

Das Ziel besteht darin, Performance-Messungen automatisiert durchzuführen mit automatischer Aufzeichnung und Auswertung systemspezifischer Performance-, Last- und Konfigurationsdaten sowie Korrelation der Messergebnisse (vgl. u. PERMEV).

Steuerung der Auswertung und Aufbereitung der Messdaten

Es gibt eine Vielzahl von Mess- und Analyse-Werkzeugen, die auf unterschiedlichen Ebenen ansetzen und häufig Insellösungen darstellen. Universell einsetzbare und automatisierte Mess- und Auswerteumgebungen sind kaum vorhanden.

Besonders aufwändig und arbeitsintensiv sind die Auswertung der Messdaten und deren statistische Aufbereitung. Da es sich häufig um etliche GigaBytes Messdaten handelt, müssen Auswertung und statistische Aufbereitung, aber auch die Erstellung von Präsentations-Tabellen und -Diagrammen voll automatisiert ablaufen. Dabei muss einerseits die automatische Erstellung von Standarddiagrammen möglich sein, und andererseits muss die automatische Erstellung von individuellen Diagrammen nach vorgebbaren Konfigurations-/ Filterkriterien, wie Auswertezeitraum, Mittelwert- und Extremwertberechnungen oder automatische bzw. vorgebbare Korrelationsberechnungen, gegeben sein (s.u. PERMEV).

(b) Exkurs: ARM – Application Response Measurement

Vor allem bei Geschäftsapplikationen ist es notwendig, diese ständig zu warten und zu verbessern. Zu diesem Zweck ist eine Überwachung der Applikation während des Betriebs durchzuführen.

Hierzu ist mit ARM eine API von der Computer Measurement Group entwickelt worden, mit der es möglich ist, den Source Code der zu untersuchenden Applikation zu instrumentieren.[2]

Der Anwender hat die Möglichkeit, den Datenfluss innerhalb der Applikation durch individuell definierbare Transaktionen zu untersuchen und Antwortzeiten zu messen. Um dies mit hinreichender Genauigkeit zu erfüllen, unterstützt ARM die Definition von Subtransaktionen, die über eindeutige Erkennungsmerkmale (Korrelatoren) einer Vatertransaktion zugeordnet werden können.

---

[2] Nähere Informationen zur ARM-Initiative gibt es unter http://www.cmg.org/regions/cmgarmw/index.html und http://www.opengroup.org/management/l2-arm.htm.

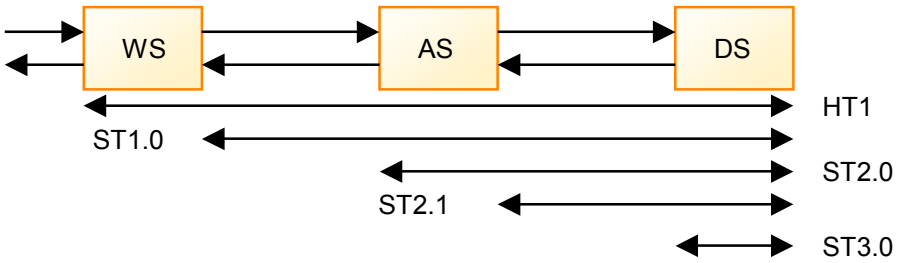
Seit der Version ARM 2.0 können Performancedaten wie Transaktionen-Durchlass, eindeutige Identifikation der Transaktionen und Transaktionsstatus ermittelt werden.

Zur Visualisierung der gesammelten Performancedaten bietet ARM standardmäßig sogenannte Logfiles an. Auch professionelle Visualisierungen mit Hilfe von Netzmanagementwerkzeugen wie HP PerfView können mit eigenen MeasureWare-Agenten erfolgen. Mit ARM 3.0 wird auch die Sprache Java unterstützt.

Stand der Technik ist, dass mit ARM mehr als nur Antwortzeiten von Transaktionen gemessen werden können. Damit verspricht ARM in Kombination mit einem Messwerkzeug (wie z.B. HP Openview GlancePlus) eine gute Option für Performance Management zu sein.

Allerdings wird für jede Applikation ein hoher Instrumentierungsaufwand gefordert, wenn die Funktionen der ARM API in den bestehenden Source-Code eingebunden werden müssen. Trotzdem wird sich ARM genau dann durchsetzen, wenn während der Entwicklung der Applikation die ARM API berücksichtigt wird. Es ist jedoch noch ein weiter Weg, bis ein „einfacher“ Einsatz von ARM in professionellen Umgebungen möglich ist.

Durch die Anwendung der ARM API und den Vergleich der gemessenen Antwortzeiten mit Dienstgütevereinbarungen kann ein Problem erkannt werden (siehe Beispiel unten). Die Problemisolation ist, nur durch die Anwendung der ARM API jedoch in nicht ausreichendem Maße zu erfüllen.



(Legende: WS = Webserver; AS=Application Server; DS = Datenbank)

Abbildung 03-11: Beispiel einer ARM-Anwendung (Datenbankabfrage)

In dieser Grafik ist eine Haupttransaktion HT1 definiert. Diese wurde um einige Subtransaktionen ST1.0 ... ST3.0 ergänzt.

Annahme: ST2.0 verstößt gegen ein SLA, ST2.1 und ST3.0 liegen innerhalb ihrer definierten SLAs.

Problembestimmung: Das Problem liegt unter Berücksichtigung der getroffenen Annahmen im Application Server. Genau hier tritt das Problem der Isolation auf. Da definierte Transaktionen oft mehrere Klassen umfassen, kann eine Problembehebung mit sehr viel Personalaufwand, und damit hohen Kosten, verbunden sein.

Problemlösung: Um das Problem genau bestimmen zu können, ist es hilfreich, wenn man genau weiß, welche Java-Methoden sehr viel Zeit verbraucht haben, um dadurch einen kritischen Pfad bei der Ausführung der Transaktion erkennen zu können. Durch das Profiling ist die Messung der verbrauchten CPU-Zeit einer Methode möglich. Da eine Verkürzung der benötigten CPU-Zeit einzelner Methoden zu einer Verringerung der gesamten Transaktionszeit führen kann, kann man hier von Problemisolation sprechen.

(c) Exkurs: Hybrid-Monitoring

Das Hybrid-Monitoring (kombiniertes Hardware- und Software-Monitoring) hat sich in der Analyse des Systemverhaltens von Rechensystemen bewährt. Die Abarbeitung verteilter Anwendungen setzt ein koordiniertes Zusammenspiel aller beteiligten Prozesse voraus. Deshalb ist für die Analyse des Systemverhaltens mittels Monitoring in derartigen Umgebungen auch ein "verteiltes Messen" erforderlich, wobei ein exakter zeitlicher Bezug zwischen den verteilt ablaufenden Prozessen herzustellen ist. Dabei sind die relevanten Messdaten dezentral zu erfassen, gegebenenfalls zu verdichten und dann zentral auszuwerten. Dafür ist ein globaler Zeitbezug erforderlich, um die verteilt erfassten Messdaten in der zeitlich richtigen Reihenfolge analysieren zu können. In räumlich verteilten und heterogenen Umgebungen ist das Software-Monitoring nur eingeschränkt möglich. Das Hybrid-Monitoring kann aber mit Hilfe einer zusätzlichen Einrichtung zur Synchronisation der verteilt angeordneten Uhren den globalen Zeitbezug herstellen.

Weitere Anforderungen an ein verteiltes Monitoring-System sind die Flexibilität in der Adaptierung an die unterschiedlichen Systeme und eine extreme Leistungsfähigkeit zur Erkennung und Erfassung sich schnell ändernder Ereignisse. Der Hardware-Monitor wird zur effizienten Erzeugung von Ereignisspuren eingesetzt, die für jedes beobachtete Ereignis eine Ereigniskennung und einen Zeitstempel (Zeitpunkt des Auftreten des Ereignisses) enthalten: Ereignisdefinition per Software, Messung und Aufzeichnung der Ereignisspur per Hardware.

(d) Test-/Messumfeld und Vorgehensweisen

Rahmenbedingungen

Zu Beginn der Analyse des Systemverhaltens müssen Zielsetzung sowie Umfang und Zeitrahmen der Untersuchung und auch die Verantwortlichkeiten festgelegt werden. Darüber hinaus sind weitere Randbedingungen zu spezifizieren: Festschreibung von quantitativen Anforderungen, wie Erwartungen des Anwenders hinsichtlich Antwortzeiten und Durchsatzraten.

Bereitstellung der Test- bzw. Messumgebung: Testbedingungen versus Realbetrieb

Messungen zur Analyse und Bewertung des Systemverhaltens werden in den Phasen ”Überwachung” und ”Analyse und Tuning” zielgerichtet im realen Produktivbetrieb durchgeführt. Die Erhebung von Performance-Werten für Modellierungs- und Lastsimulations-Parameter zur ”Planung und Prognose” ist dagegen in einem klinischen Testumfeld, wo keine Störeinflüsse das Systemverhalten beeinflussen, wesentlich einfacher und unproblematischer durchzuführen. Dafür muss dann die relevante System- und Anwendungsumgebung incl. der Datenbasis bereit gestellt werden.

Spezifizierung der Last

Sowohl für die Analyse des Systemverhaltens in den Phasen “Überwachung“ und “Analyse und Tuning“ als auch zur Ermittlung von Performancedaten für die “Planung und Prognose“ ist eine ausführliche Lastspezifikation erforderlich. Neben der Festlegung von Normal- und Spitzenlasten sind dabei auch Zusatz- und Hintergrundlasten zu berücksichtigen.

Bei der Ermittlung von Performance-Daten für Modellierungsparameter ist die Lastspezifikation auf Basis von Arbeitsplatztypen (AP-Typen) zur Charakterisierung des Benutzerverhaltens verschiedener Anwendungsbereiche und Benutzergruppen anzustreben. Auswahlkriterien sind dabei:

- Differenzierung nach deutlich abweichendem Benutzerverhalten
- Auswahl relevanter Benutzergruppen mit erwartungsgemäß nicht zu vernachlässigender Belastung des Gesamtsystems

Weiterhin müssen die Anzahl der aktiven Arbeitsplätze und die Verhältniszahlen der AP-Typen zur Angabe der zu vermessenden Anzahl von Arbeitsplätzen und der Verteilung der spezifizierten AP-Typen auf diese festgelegt werden.

Die Ermittlung des Ressourcenbedarfs der GP bzw. IP-Prozesse muss AP-Typ-spezifisch auf der Ebene von Anwendungsvorgängen mit Transaktionsbeschreibungen erfolgen. Dabei handelt es sich auf logischer Ebene um Vorgänge zur groben Beschreibung der Tätigkeiten für die ausgewählten AP-Typen, wobei repräsentative und zeit- und belastungskritische Vorgänge vorrangig zu betrachten sind. Die Transaktionsbeschreibungen dienen zur genauen Beschreibung der Vorgänge und müssen ggf. in mehrere Transaktionsschritte zerlegt werden, die dann als Einzel- oder Teiltransaktionen vermessen werden.

Ermittlung des transaktionsspezifischen Ressourcenbedarfs

Einsatz von Messwerkzeugen

Für jede Messreihe sind die folgenden Analysen durchzuführen:

@ Ermittlung der transaktionsspezifischen Antwortzeiten

@ Überprüfung und Ermittlung der Auslastungsgrade der Systemressourcen

@ Ermittlung der spezifischen Ressourcen-Verbräuche der stationären und temporären Prozesse

Dafür sind problemadäquate Messverfahren und -werkzeuge einzusetzen.

Einplatzmessungen

Transaktionsanalysen sind am besten als Einplatzmessungen im Testumfeld durchzuführen. Dabei wird folgendermaßen vorgegangen:

@ Bestandsaufnahme der Dienstgüte mittels Ermittlung der transaktionsspezifischen Antwortzeiten

@ Ermittlung der Grundlast, d.h. die Prozesse der ausgewählten Transaktionen sind ablaufbereit, aber inaktiv

@ Ermittlung der transaktionsspezifischen Ressourcen-Verbräuche an einem aktiven Arbeitsplatz

Mehrplatzmessungen

Zur Ermittlung der gegenseitigen Beeinflussung durch die parallel ablaufenden Prozesse aller ausgewählten Transaktionen müssen Mehrplatzmessungen durchgeführt werden. Dabei sind folgende Aspekte zu beachten:

@ Nachbildung der Volllast: Um den Einfluss des unterschiedlichen und manchmal auch fehlerhaften Benutzerverhaltens zu eliminieren, werden anstelle der realen Benutzer Lasttreibersysteme eingesetzt (vgl. Testautomatisierung). Falls keine realen Applikati-

onen eingesetzt werden können, muss auf die Lastsimulation (s.o.) zurückgegriffen werden.

@ Ermittlung der Systembelastung: Da hierfür i.d.R. unterschiedliche Messwerkzeuge eingesetzt werden und diese unsynchronisiert ablaufen und damit auch unkorrelierte Messdaten liefern, müssen diese Werkzeuge zentral gesteuert werden (vgl. Messautomatisierung). Dafür sind automatisierte Mess- und Auswerteumgebungen gut geeignet (s.u. PERMEV).

Bei allem Messungen muss auf folgendes geachtet werden:

@ In den meisten Fällen werden Messwerkzeuge eingesetzt, die im Sampling-Verfahren arbeiten, d.h. dass es sich bei den Messungen um "Zufallsexperimente" handelt. Deshalb müssen bei allen Messungen Replikationen durchgeführt werden, um Ausreißer eliminieren zu können.

@ Ebenso muss auf die Einhaltung des "eingeschwungenen Zustandes" geachtet werden. Es müssen Vorlauf- und Nachlaufphasen eingehalten werden, und der Messzeitraum sollte größer als die fünffache Zykluszeit sein.

@ Verlässlichkeit und Konsistenz der Messdaten: Dazu müssen alle Fehlersituationen und Ausfälle protokolliert werden, um über die Gültigkeit der Messung entscheiden zu können.

(e) Test- und Messautomatisierung

Automatische Lasterzeugung

Zur Analyse und Bewertung des Gesamtsystemverhaltens ist die Einbeziehung einer definierten Last unabdingbar. Die Umsetzung der realen Last erfolgt in Form von so genannten Lastprofilen. Dabei stehen in diesem Umfeld zwei Ansätze zur Verfügung, die auch kombinierbar sind. Zum einen kann die reale Last in synthetische Lastprofile umgesetzt werden, d.h. die Anwenderlast wird mittels vorgegebener Funktionen und Aufträge realitätsgetreu nachgebildet und unabhängig von der realen Applikation ausgeführt. Im anderen Fall erfolgt die Umsetzung der Anwenderlast in reale Lastprofile, d.h. es werden die Eingaben und das Verhalten der Benutzer im realen Applikationsumfeld synthetisiert, worauf sich im Folgenden die Test Automatisierung bezieht.

Anforderungen

Damit Messungen auf Basis von realen Lastprofilen in vergleichbaren Gesamtsystemen auch vergleichbare Ergebnisse liefern, sind bestimmte grundlegende Anforderungen zu erfüllen:

@ Generierbarkeit: Die vorgegebenen Anwendungsfälle müssen die Umsetzung in erzeugbare Lastprofile unterstützen, d.h. die Last muss aus einer zusammenhängenden Abfolge von Benutzereingaben bestehen. Dieses Kriterium ist auch Voraussetzung, um ein reales Benutzerverhalten nachzuvollziehen.

@ Reproduzierbarkeit: Die Lastprofile müssen wiederholt ausführbar sein, um die Last für bestimmte Zeitspannen erzeugen zu können.

@ Überprüfbarkeit: Während der Erzeugung der Last ist die Aufnahme gewisser Messdaten (z.B. Durchsätze, Antwortzeiten) an den Erzeugungssystemen notwendig. Diese dienen einerseits zur Kontrolle der Einhaltung der vorgegebenen Lastdefinition und anderseits zur Quantifizierung der Auswirkungen der erzeugten Last auf das Verhalten des Gesamtsystems.

Zur statistischen Absicherung der verschiedenen Messfälle sollte die Wiederholung der Lastzyklen beliebig oft erfolgen können. So ist die Reproduzierbarkeit der Ergebnisse gewährleistet. Damit ist die Notwendigkeit einer automatischen Lasterzeugung gegeben.

Die Durchführung von Messungen mit realem Personal in einem Umfeld von vielen Clientsystemen scheitert an deren Verfügbarkeit und den damit verbundenen Kosten. Aber auch bei einer kleinen Anzahl von Clientsystemen ist die händische Lasterzeugung mit Unwägbarkeiten verknüpft, da durch

@ fehlerhafte Eingaben

@ unkontrolliertes Zeitverhalten bei den Eingaben

@ unterschiedliche Arbeitsweisen verschiedener Benutzer

keine kontrolliert ablaufende Lasterzeugung stattfindet.

Weiterhin wird die Anforderung der Variabilität gestellt: Durch Parametrisierung muss eine einfache und schnelle Ausführung verschiedener Lastprofile mit unterschiedlichen Durchsätzen für beliebige Zeitdauern erfolgen können. Bei der Beteiligung vieler Last erzeugender Clients ist eine zentrale Steuerungsinstanz einzusetzen, welche die Messungen vorbereitet, durchführt und überwacht sowie nachbereitet (vgl. Messautomatisierung).

Neben der Performance-Messung wird die automatische Lasterzeugung auch zur Testautomatisierung in der Quality Assurance eingesetzt. In diesem Zusammenhang sind die folgenden Punkte besonders wichtig:

@ Fehler-Handling

@ Vollständigkeit

@ Testabdeckung

Jede Performance-Messung eines Gesamtsystems beinhaltet gleichzeitig einen funktionalen Test des Systems. Somit ist die zeitgenaue Protokollierung des Fehlverhaltens für die Interpretation der Ergebnisdaten erforderlich.

Fazit: Mit der automatischen Lasterzeugung wird erreicht, dass definierte und reproduzierbare Lastprofile ohne großen personellen Aufwand zur Analyse und Bestimmung des Verhalten des zu betrachtenden Gesamtsystems erzeugt werden können.

Durchführung von automatisierten Last- und Performance-Tests

Aufgabenstellung

In vernetzten IT-Umgebungen werden Last- und Performance-Tests zweckdienlich mit einer großen Anzahl (>> 100) von Clients durchgeführt, um so möglichst realitätsnah und vielseitig Lasten erzeugen zu können. Insbesondere ist die Einbeziehung des Netzwerkes bei solchen Tests ein nicht zu vernachlässigender Aspekt, dem nur mit der Beteiligung von sehr vielen Netzknoten (Clients) genüge getan wird. Erst die Möglichkeit, an vielen Clients Lasten zu erzeugen, die alle nahezu gleichzeitig oder auch nur sporadisch auf das Netzwerk und die Server-Systeme zugreifen, können Funktionalität und Performance des vorhandenen Netzes und somit das Verhalten des Gesamtsystems entsprechend analysiert und bewertet werden. Die Durchführung von Last- und Performance-Tests mit vielen Client-PCs macht es erforderlich, Mechanismen zur definierten Ansteuerung und Überwachung der Clients zur Verfügung zu stellen.

Hierbei sind zu nennen:

@ Verteilung der Skripte, Programme und Eingabe- oder Konfigurationsdateien

@ Zuteilung bestimmter Rollen in dem Last- oder Performance-Test

@ Auswahl und Parametrisierung verschiedener Lastfälle

@ gezielte Startaufforderung zur Lasterzeugung an die PCs

@ Einsammeln von erzeugten Ausgabedateien und erhobener Messergebnisse von den PCs

@ Überprüfung der korrekten Lastdurchführung

@ Einsammeln und Auswerten der Messdaten

Des Weiteren müssen diese Mechanismen automatisch ablaufen, denn es ist sehr aufwändig, händisch alle Clients derart vorzubereiten, dass sie an dem Test partizipieren können. Andererseits lassen sich ohne diesen Automatismus gewisse Tests gar nicht oder nur mit

sehr hohem Personalaufwand realisieren. Dazu ist ein Steuerungsprogramm erforderlich, das die Durchführung der Last- und Performance-Tests von einer zentralen Instanz aus steuert.

Funktionalitäten des Steuerungsprogramms

@ Es steht eine Synchronisierungsmöglichkeit zur Verfügung.

@ Kommandos werden an den Client- und an den Serversystemen von der zentralen Steuerungsinstanz initiiert ausgeführt.

@ Dateien sind zwischen der zentralen Steuerungsinstanz und den Clients sowie Servern in beide Richtungen übertragbar.

Zentrale Steuerungsinstanz und Ablaufumgebung

Die Realisierung basiert auf einem im Rahmen von MAPKIT entwickelten Werkzeug, mit dem Zeichenketten zentral von einem System über Netzwerk zu anderen Systemen übertragen werden können. Dieses Werkzeug beinhaltet einen Sender- und einen Empfängerteil.

Schematischer Ablauf eines Last- und Performance-Tests

1. Vorbereitung

@ Erstellen einer Liste mit am Lasttest beteiligten Clients und einer Liste der beteiligten Server-Systeme

@ Bereitstellen der für den Lasttest notwendigen Dateien und Programme - insbesondere die Lasterzeugungsumgebung

@ Übertragung der Kommandofolge an Server und Clients zum Kopieren der jeweils notwendigen Dateien

@ Übertragung der Kommandofolge an Server und Clients zur Durchführung server- und clientspezifischer Maßnahmen (z.B. Anmeldung, Rücksetzen der Testumgebung auf einen definierten Stand, ...)

2. Messung

@ Übertragung der Kommandofolge an die Clients und ggfs. Server zum Start der Lasterzeugung (Verzögerungszeiten möglich)

@ Übertragung der Kommandofolge an Server und ggfs Clients zum Start der Mess-Werkzeuge

@ Abwarten der Messdauer

3. Nachbereitung

- Überprüfung aller beteiligten Systeme auf Beendigung des Tests
- Übertragung der Kommandofolge an Server und Clients zum Einsammeln der Ergebnis- und Protokolldateien
- Vorauswertung der Ergebnisdateien an der Steuerungsinstanz

KAPITEL 04 WERKZEUGE

R. BORDEWISCH, C. FLUES, R. GRABAU, J. HINTELMANN,
K. HIRSCH, F.-J. STEWING

04.01 Mess- und Monitoring-Werkzeuge

Im Folgenden werden Mess- und Monitoring Werkzeuge vorgestellt, die aus den praktischen Erfahrungen im MAPKIT-Projekt entstanden sind bzw. erprobt wurden. Die folgende Aufstellung ist untergliedert in "Werkzeuge zur Ermittlung der Dienstgüte", "Automatisierte Mess- und Auswerteumgebungen" und "Messwerkzeuge zur Analyse in vernetzten Umgebungen" und erhebt keinen Anspruch auf Vollständigkeit.

(a) Werkzeuge zur Ermittlung der Dienstgüte

QoS (Quality of Service) in Netzwerken umfasst neben der Verfügbarkeit der IT-Infrastruktur auch die Sicherstellung zugesagter Antwortzeiten für einzelne oder komplette Applikationsfunktionen. Aus Applikationssicht müssen die parallel genutzte Ressource Netzwerk und die darauf laufenden Anwendungen (File-Sharing, Datenbank, zentrale Services, ...) leistungsmäßig betrachtet und beurteilt werden können. Die Festschreibung von Service-Leveln erfolgt in Angaben zu Performance-Kriterien und Sollwerten. Die Betriebsstrategie „Reaktion" bedeutet Warten auf Verletzung des Service-Level-Agreements und reicht in heutigen Anwendungsszenarien bei weitem nicht mehr aus. Ziel muss vielmehr ein „proaktives Management" sein, das Störungen im Ansatz erkennt und behandelt, bevor sie vom Anwender als Störung empfunden werden.

Agententechnologie

System- und Netzwerkmanagementsysteme (sog. Frameworks) bieten im Rahmen des Resource and Performance Managements Werkzeuge zur Ermittlung und Überwachung der Dienstgüte (z.B. Antwortzeiten) an, die i.d.R. auf der Agententechnologie basieren. Auf den verteilten und zu vermessenden Netzknoten/Systemen werden Kollektoren installiert, die im Betriebssystem, in Datenbanken und Applikationen (z.B. SAP R/3) Performance-Daten erheben, die dann ausgewertet und grafisch aufbereitet werden. So können auch auf Client-PCs Kollektoren installiert werden, die dann die Applikation instrumentieren, die Antwortzeiten ermitteln und so die Dienstgüte überwachen. Dieser Ansatz weist aber auch einige teilweise gravierende Nachteile auf:

- @ Frameworks und Spezialwerkzeuge sind nicht immer verfügbar
- @ Problematik der hohen Lizenzkosten
- @ Oft fehlende Erweiterbarkeit der Mess- und Analysewerkzeuge
- @ Software-Sensoren sind nicht immer leicht implementierbar

Deshalb wurde im Rahmen von MAPKIT das Proaktive Kapazitätsmanagement auf Basis des Referenz-PC entwickelt und erprobt.

Referenz-PC (Siemens)

Der Referenz-PC bietet die Möglichkeit, auf performancekritische Antwortzeiten schnell zu reagieren bzw. diese im Vorfeld durch eine Trendanalyse oder Prognose erkennen und abwehren oder abschwächen zu können. Die Software wird auf Windows-Clients eingesetzt und simuliert einen realen Benutzer an der Schnittstelle der grafischen Oberfläche (= Bildschirm/Tastatur), der auf dem jeweiligen Server definierte Transaktionen durchführt.

Die wiederkehrende Ermittlung der Antwortzeiten von definierten Transaktionen ermöglicht Vergleiche zu vorherigen Messungen, die Rückschlüsse auf das allgemeine Serververhalten bzgl. Ressourcenhandling und Engpass-Situationen ermöglichen. Dabei werden in der realen Infrastruktur auf einem exklusiv verfügbaren Client-PC business-kritische Applikationen zum Ablauf gebracht, wobei der Endbenutzer mittels eines automatischen Testtreibers simuliert wird.

Die realen Eingaben (Tastatur-Eingaben, Mausklicks) werden mit Hilfe von Visual Tests-Skripten nachgebildet, und es werden so auch die relevanten Antwortzeiten ohne Modifikation der zu überwachenden Applikation und ohne Rückkoppelungseffekte auf das Gesamtsystem ermittelt. Es ist erstrebenswert, den Referenz-PC sowohl im zentralen LAN als auch in der dezentralen Filiale zu betreiben, um so Rückschlüsse auf den Einfluss eines WAN auf das Verhalten des Gesamtsystems ziehen zu können (vgl. Abbildung 04-1).

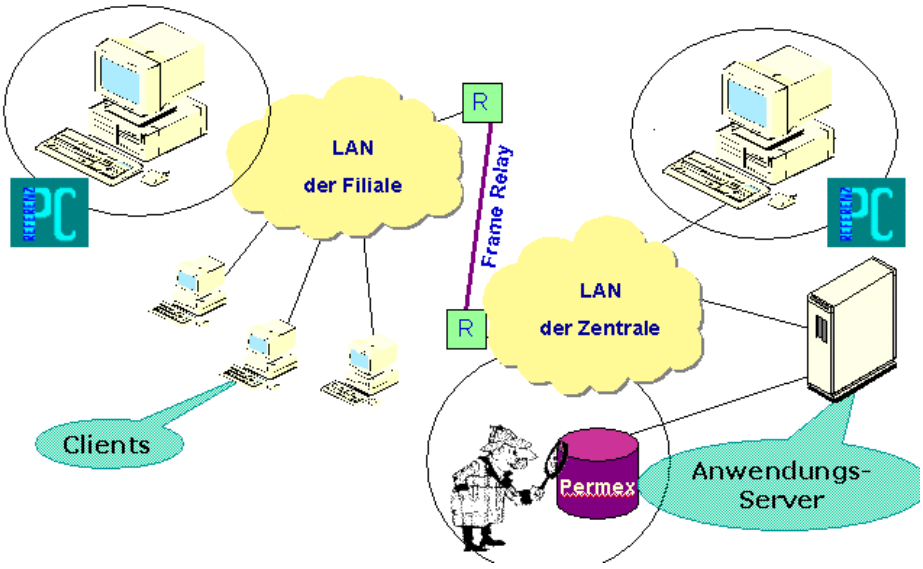


Abbildung 04-1: Szenario mit Referenz-PC

Proaktives Kapazitätsmanagement (Siemens)

Heutige Netzwerkmanagement-Tools basieren i.d.R. auf der Überwachung der Auslastungsgrade der System- und Netzwerk-Komponenten und ermöglichen nur sehr eingeschränkt Detail- und Ursachenanalysen des Systemverhaltens. Ein bereits früher seitens Siemens Business Services verfolgter Ansatz ist die Überwachung und Analyse der Server-Ressourcen mittels PERMEV. Ziel des Proaktiven Kapazitätsmanagements ist es, bei Verletzung der Dienstgüte gezielt Detail- und Ursachenanalysen hinsichtlich des Verhaltens der Systemkomponentenvorzunehmen, um den IT- und Applikationsbetreiber in die Lage zu versetzen, frühzeitig Maßnahmen einzuleiten, bevor sich ein Anwender beschwert.

Das Proaktive Kapazitätsmanagement in Form der Koppelung des Referenz-PC mit der bewährten PERMEV-Langzeitüberwachung bietet folgende Vorteile:

@ Die PERMEV-Langzeitüberwachung ermittelt die Ressourcenauslastung am Server und stellt eine Trendanalyse über einen vorzugebenden Zeitraum bereit. Die Messintervalle sind dabei relativ groß gewählt, um das System nicht unnötig zu belasten.

@ Am Referenz-PC werden mittels eines Software-Treibers (Simulation eines realen Benutzers an der grafischen Oberfläche) business-kritische Transaktionen in festzulegenden Abständen durchgeführt und deren Antwortzeit gemäß DIN 66273 ermittelt

und protokolliert. Für jede Transaktion kann ein Schwellwert definiert werden (Service Level Agreement (SLA) zwischen IT-Betreiber und Fachabteilung).

@ Bei Schwellwertüberschreitungen am Client werden neben einem Alert an den Administrator durch PERMEV zusätzliche Messdaten erhoben. Durch die Korrelation der Messdaten wird eine gezielte Ursachenanalyse am Server mit den bereits vorhandenen Messdaten ermöglicht. Gleichzeitig werden für einen kurzen Zeitraum die Client-Transaktionen in verkürztem Abstand durchgeführt, um das Antwortzeitverhalten bis zu einer „Normalisierung" detailliert zu verfolgen.

(b) Automatisierte Mess- und Auswerteumgebungen

PERMEX (Siemens)

Das UNIX-Betriebssystem bietet standardmäßig eine Reihe von Software-Monitoring-Werkzeugen an. Der bekannteste Vertreter ist der sar (System Activity Reporter), der eine breite und vielfältige Performance-Analyse der wichtigsten Systemkomponenten und -funktionen ermöglicht. Das Kommando netstat liefert Performance-Daten der angeschlossenen LANs. Weiterhin wird sehr häufig ps (process state) und acct (accounting) zur Ermittlung prozessspezifischer Daten eingesetzt. Das Kommando dkstat erlaubt eine detaillierte Betrachtung des Plattenzugriffsverhaltens und ebenso kann mit Hilfe der Kommandos etherstat und atmstat eine tiefergehende Analyse bestimmter Netztopologien erfolgen. Zur Analyse der Last auf der Ebene des Netzwerk-Betriebssystems LM/X (LAN Manager for UNIX) bzw. AS/X (Advanced Server for UNIX) wird lmstat und bei NFS nfsstat eingesetzt. Diese Monitore werden unterschiedlich gestartet, laufen unsynchronisiert ab und liefern unterschiedliche, unkorrelierte Messdaten. Um die Anwendung dieser verschiedenen Werkzeuge zu vereinfachen, um deren Aufrufe zu standardisieren und zu synchronisieren sowie die Messergebnisse zu korrelieren und um insbesondere den Aufwand für die Durchführung von Performance-Messungen und deren Auswertung drastisch zu reduzieren, wurde die voll automatisierte Monitoring- und Auswerte-Umgebung PERMEX (Performance Monitoring and Evaluation Environment for Reliant UNIX) realisiert, wobei Monitoring- und Auswerte-Umgebung getrennt eingesetzt werden.

Das in PERMEX enthaltene Monitoring-Skript mon wird zur mittel- und langfristigen Performance-Messung auf einem RELIANT UNIX-System eingesetzt. Zielrichtungen sind einerseits die Langzeitüberwachung zur Trendanalyse und andererseits die gezielte Ursachen- und Engpassanalyse. Dazu wird die Monitoring-Umgebung konfiguriert, auf das zu vermessende System kopiert und aktiviert. Das Skript erlaubt den konfigurierbaren Einsatz der unterschiedlichen UNIX-Messwerkzeuge und die Aufnahme von relevanten Systeminformationen. Die erhobenen Daten werden komprimiert in einer Ergebnisdatei archiviert.

Eine Erweiterung des Skripts um zusätzliche Messwerkzeuge (Traces oder Statistiken) ist schnell und einfach möglich.

Nach Erhalt der per mon-Skript erhobenen Messdaten werden diese offline mittels Auswerte-Skript eval ausgewertet und in Tabellen zusammengefasst, die z.B. durch EXCEL weiterverarbeitet werden können. Neben einer Standardauswertung mit den wesentlichen Performance-Kenngrößen des vermessenen Systems kann die Zusammensetzung dieser Ergebnis-Tabellen vielfältig individuell konfiguriert werden; u. a. ist es möglich Operationen wie Mittelwert-, Maximum- oder Summenbildung über verschiedene Performance-Kenngrößen zu definieren.

PERMEX ermöglicht

- die vereinfachte Handhabung unterschiedlicher Messwerkzeuge,
- den Remote-Einsatz,
- die Korrelation der Messergebnisse,
- die einheitliche, aussagekräftige Systemübersicht,
- die drastische Reduzierung des Aufwands für Messdurchführung und Auswertung und
- hat sich in etlichen Einsätzen bestens bewährt.

PERMENT (Siemens)

Sowohl bei der Langzeitüberwachung als auch zur Lokalisierung und Analyse von Performance-Engpässen in realen Umgebungen mit Reliant UNIX-Server-Systemen wurden mit der automatisierten Mess- und Auswerteumgebung sehr gute Erfahrungen gesammelt. Aufgrund des Wandels der IT-Infrastruktur in heutigen Client/Server-Anwendungen zu MS Windows NT-basierten Server-Systemen wurde für MS Windows NT PERMENT (Performance Measurement and Evaluation Environment for MS Windows NT) mit folgenden Eigenschaften realisiert:

- weitgehend bedienerlose, einfache Erhebung von Systeminformationen und Performance-Daten
- schnelle Konfigurierbarkeit nach Bedürfnissen der Kunden
- Unterstützung bei Interpretation von Messergebnissen und Erstellung von Präsentationsunterlagen
- eine Mess- und Auswerteumgebung unter MS Windows

Wie auch PERMEX umfasst PERMENT eine Monitoring- und eine Auswerte-Umgebung, welche getrennt eingesetzt werden. Die Basis für die in PERMENT enthaltene Monitoring-Umgebung bildet der im NT-Resource Kit verfügbare Perf Data Log Service. Dieser Hintergrunddienst zeichnet mittel- oder langfristig intervallgesteuert Performance-Daten des betrachteten Systems auf. Zusätzlich werden durch spezielle Programme der Monitoring-Umgebung relevante Systeminformationen zu Beginn und am Ende einer Messung erhoben. Zur Durchführung von Messungen wird die Monitoring-Umgebung konfiguriert, auf das zu vermessende System kopiert, und aktiviert. Die erhobenen Daten werden pro Messung komprimiert in einer Ergebnisdatei archiviert.

In der vorliegenden Version wird für die Auswertung und die Ergebnisaufbereitung der Messdaten die Auswerte-Umgebung von PERMEX genutzt. Demzufolge werden nach Erhalt der erhobenen Messdaten diese mit Hilfe des PERMEX-Auswerte-Skripts ausgewertet und in Tabellen zusammengefasst, die z.B. durch EXCEL weiterverarbeitet werden können.

PERMEV (Siemens)

Basierend auf dem PERMEX-Konzept wurde für Systeme mit UNIX-basiertem Betriebssystem die Mess- und Auswerteumgebung PERMEV realisiert. Die derzeitige Version ist lauffähig unter den UNIX-Derivaten Reliant UNIX, SOLARIS, LINUX und HP_UX.

PERMEV (Performance Monitoring and Evaluation Environment) setzt sich zusammen aus den Komponenten

- Mess-System ‚monitor'
- Auswertesystem ‚evaluation'
- Präsentationssystem ‚statistics'

Das Mess-System ‚monitor' sammelt Performance-Daten auf dem zu vermessenden System. Es handelt sich dabei um Daten über die Belegung der Systemressourcen wie CPU, Hauptspeicher, Festplatten und Netzzugang sowie prozessspezifische Daten und Oracle-Statistiken. Die Erweiterung um R/3-Statistiken ist in Arbeit. Die erhobenen Messdaten werden innerhalb der Auswerteumgebung durch die Komponente ‚evaluation' verdichtet und tabellarisch zusammengefasst, wobei das Auswertesystem hinsichtlich der auszuwertenden Performance-Daten variabel konfigurierbar ist. Die durch das Auswertesystem erzeugten Ergebnisdaten werden nachfolgend durch die Komponente ‚statistics' statistisch weiter ausgewertet und aufbereitet. Hierzu gehört die Ermittlung von Mittel- und Maximalwerten, die Berechnung von Korrelationskoeffizienten sowie die grafische Präsentation.

Im Mess-System ‚monitor' werden mit Hilfe des SHELL-Skripts permon und der - je nach Betriebssystem - verfügbaren Standardmesstools die Performance-Daten gesammelt und als Rohdaten in einzelnen Dateien abgelegt. Die Konfigurationsdatei permon.cfg definiert, welche Tools aufgerufen werden. Am Schluss einer Messung werden diese Daten in der Archivdatei komprimiert verpackt.

Im Auswertesystem ‚evaluation' werden mit Hilfe des PERL-Skripts eval.pl und weiterer PERL-Moduln aus den Rohdaten über eine Vorauswertung und evtl. eine anschließende weitere Datenverknüpfung Ergebnistabellen erzeugt. Die Vorauswertung wird gesteuert durch die Konfigurationsdatei preeval.cfg. Abschließend wird eine tabellarische Ergebnisübersicht generiert, deren Umfang über die Konfigurationsdatei eval.cfg variabel spezifizierbar ist. In den Ergebnistabellen und der Ergebnisübersicht sind die einzelnen Performance-Daten zeitlich den jeweiligen Messintervallen zugeordnet. Im Präsentationssystem ‚statistics' erfolgt mit Hilfe eines Excel-Makros eine weitere u.a. grafische Aufbereitung der Ergebnisübersicht. Die Konfigurationsdatei statistics.cfg legt Form und Umfang der dabei resultierenden Tabellen und Diagramme fest.

(c) Messwerkzeuge zur Analyse in vernetzten Umgebungen

Software-Netzwerk-Analysator: dp_commat (Siemens)

Das Monitoring der Netzwerk-Aktivitäten erfolgt meist durch den Einsatz von Netzwerk-Analysatoren. Bekannte Vertreter sind Sniffer von Network Associates oder Domino von Wandel & Goltermann. Im Rahmen des Netzwerk-Managements kommen primär SNMP-basierte LAN-Probes und Router/Bridge/Switch-Statistiken zur permanenten Netzüberwachung zum Einsatz, die funktionale Aspekte des Netzwerks abdecken. Diese Verfahren sind aber nur bedingt einsetzbar, um Basisdaten z.B. für eine anstehende Netzwerk-Dimensionierung oder Server-Farm-Konsolidierung zu ermitteln.

Zur Aufnahme der für Engpassanalysen und Netz-Dimensionierungen erforderlichen Basisdaten wurde daher das Linux-basierte Werkzeug dp_commat entwickelt. Dieses Tool überwacht ähnlich einer LAN-Probe den gesamten Netzverkehr eines Segments und liefert durch die Interpretation der Pakete detaillierte Informationen zur Beantwortung der Fragen:

- Wer kommuniziert mit wem?
- Welche Paketgrößen treten wie häufig auf?
- Zu welchen Zeiten traten welche Netzlasten auf?
- Welche Protokolle sind zu welchen Anteilen an der gesamten Netzlast beteiligt?

Diese Informationen werden absolut passiv ermittelt und können auch über Zeiträume von mehreren Tagen erhoben werden. Da das Tool unter Linux abläuft, ist keine teuere Spezial-Hardware erforderlich, sondern ein Standard-PC erfüllt fast alle Bedürfnisse. Der Einsatz eines Spezialisten vor Ort kann durch die einfache Handhabung somit meist entfallen.

Die Auswertung der durch dp_commat erhobenen Daten erfolgt werkzeuggestützt. Als Ergebnisse werden abhängig von den Anforderungen geliefert:

- qualifizierte und realitätsgetreue Netzwerk-Dimensionierungen
- Identifikation auch von sporadischen Netzwerk- oder Netzwerkknoten-Problemen
- Hinweise auf Server-Engpässe und evtl. hieraus resultierende sinnvolle Verlagerungen von Diensten, welche das Netzwerk nutzen
- Hinweise auf Netzwerk-Engpässe und sinnvolle Umstrukturierungen

Mess- und Analysetool „Application Expert“ (Compuware)

Das Werkzeug Application Expert (AE) eignet sich gut zur Performance-Optimierung beim Betrieb von mehrstufig vernetzten Client/Server-Anwendungen. AE ist ein Produkt der Firma Compuware (früher Optimal Networks). Haupteinsatzfelder sind die Analyse von verteilten Anwendungen in Bezug auf deren Laufzeitverhalten und gleichzeitig das Aufzeigen von konkretem Tuningpotenzial.

Unmittelbarer Nutzen und Erkenntnisse beim Einsatz von AE lassen sich wie folgt kategorisieren:

- Nachweis der Gesamtlaufzeit eines konkreten Geschäftsprozesses aus Sicht des End-Users
- Exakte Zuordnung der Zeitanteile zu den involvierten Hardware-Komponenten (Client, Netz, Applikationsserver, Datenbankserver) und Erkennen zeitlicher Synchronisationsmuster bei verteilt ablaufenden Transaktionen
- Nachweis des Datenverkehrsaufkommens zwischen den beteiligten Komponenten
- Gezieltes Erkennen von Tuningbedarf und -möglichkeiten bereits vor einem produktiven Einsatz von C/S-Anwendungen aufgrund komfortabler Facilities zur Top-down-Analyse
- Prognose zum Laufzeitverhalten von Geschäftsprozessen bei variierender Netzbandbreite
- Automatisierte, ansprechende grafische und tabellarische Aufbereitung von Ergebnissen

AE ist auf allen Windows-Plattformen einsetzbar. Die überwachten HW-Objekte können von beliebiger Art sein. Die Erfassung der Messdaten erfolgt über das Netzwerk-Interface eines PCs oder Notebooks sowie durch den Import von Trace-Files von Protokolltestern oder Netzwerkmanagement-Tools. Soweit grundlegende Kenntnisse im Bereich Performance und Netzwerk-Monitoring vorliegen, ist der effiziente Umgang mit AE relativ leicht zu erlernen.

Die Instrumentierung von AE kann man für konkrete Untersuchungszwecke unterschiedlich anpassen. Durch entsprechendes Setzen von Filtern lässt sich gezielt der Datenverkehr zwischen 2 einzelnen IP-Adressen, mehreren Paaren von IP-Adressen oder auch zwischen einer IP-Adresse und allen anderen Adressen, die mit dieser Daten austauschen, aufzeichnen. Es ist aber auch möglich, den gesamten Datenverkehr zu erfassen. Je nach Art der im Monitorsystem eingebauten Netzwerkkarte ist es so möglich, im Ethernet, Fast-Ethernet, FDDI oder Token Ring zu messen.

Durch spezielle Importfeatures ist es möglich Messungen von Protokolltestern (z.B. Wandel und Goltermann) oder Netzwerkmanagementsystemen (z.B. HP Openview) zu analysieren. Neben den umgehend verfügbaren Standardergebnissen erlauben es die oben dargestellten Diagnosemöglichkeiten somit, das Laufzeitverhalten von Geschäftsprozessen topdown zu untersuchen und ggf. einzelne Aufrufe auf Applikationsebene als besonders ressourcen- und zeitintensiv zu identifizieren. Darüber hinaus ist es möglich, mit den Funktionen

- Response Time Predictor, das Laufzeitverhalten einer untersuchten Transaktion bei verschiedenen Netzbandbreiten analytisch zu berechnen.
- Bandwidth Estimator, die zwischen 2 IP-Adressen verfügbare Netzbandbreite experimentell zu ermitteln.
- Latency Finder, die Signallaufzeit zwischen 2 IP-Adressen zu bestimmen.
- Comparison Report, identische, mehrfach untersuchte Abläufe/Transaktionen automatisch zu vergleichen.

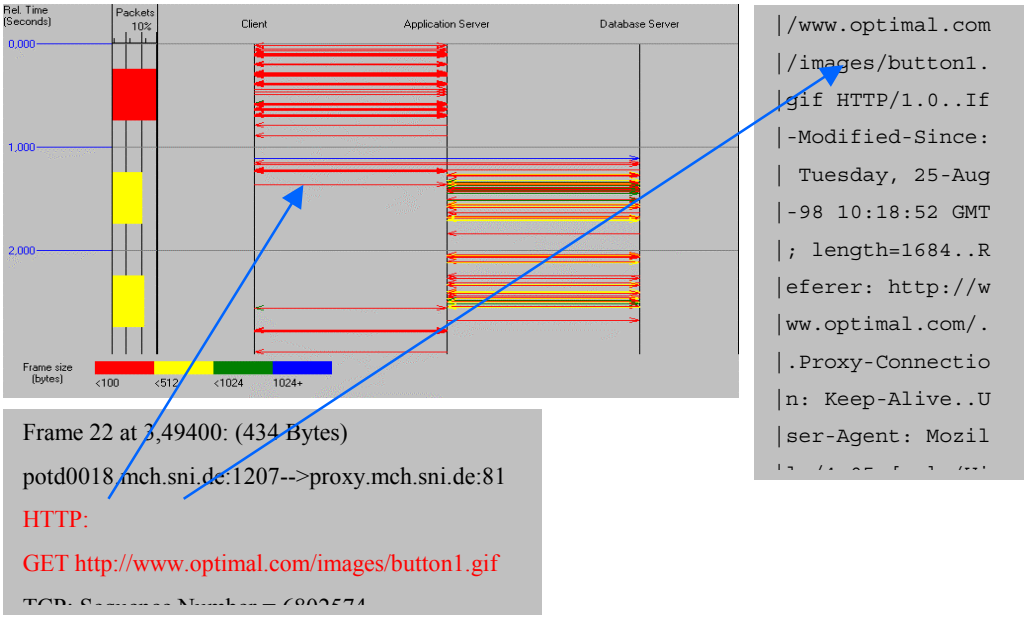


Abbildung 04-2: Kommunikationsvorgänge, aufgezeichnet mit dem Application Expert

Das Beispiel zeigt die für einen Geschäftsprozess anfallenden Kommunikationsvorgänge zwischen einem Client, Applikationsserver und Datenbankserver im zeitlichen Verlauf. Im Kasten rechts erkennt man den Inhalt eines einzelnen Datenpakets, während im unteren Kasten der Inhalt dieses Datenpakets auf das HTTP-Protokoll reduziert wird. Neben dem HTTP-Protokoll können noch eine Reihe weiterer Protokolle, wie z.B. SQL, FTP, SMTP u.a. analog decodiert werden.

Abgrenzung zu anderen Messverfahren:

Der Einsatzschwerpunkt von AE ist weniger das Aufzeichnen des gesamten im Multiuser-Betrieb von C/S-Applikationen erzeugten Netzverkehrs, sondern vielmehr der Mitschnitt und die Analyse in Bezug auf einzelne, definierte Geschäftsprozesse.

Idealerweise sollten die mit AE in einer C/S-Umgebung überwachten Objekte innerhalb eines Netzsegments liegen. AE protokolliert den Datenverkehr zwischen den beim Ablauf eines Geschäftsprozesses beteiligten Komponenten. An den überwachten Objekten - Hard-ware und Software - sind aufgrund dieser Art der Messdatenerfassung keinerlei Anpassun-

gen vorzunehmen. Es ergeben sich somit auch keine durch Mess-Overhead verursachten Rückkoppelungseffekte auf das beobachtete Ablaufverhalten.

Allerdings liefert alleine diese Art des Monitorings keine Erkenntnisse über den aktuellen Auslastungszustand der beobachteten Systeme (z.B. Clients oder Server). Man stellt aber die direkte Auswirkung der auf diesen Komponenten verfügbaren Systemleistung auf das Laufzeitverhalten fest. Je nach Untersuchungsziel ist es dann auch möglich, die AE-Ergebnisse mit Messdaten zu korrelieren, die zeitgleich über andere Schnittstellen (z.B. in einem Server) erfasst wurden.

ZM4 & DA30 (Siemens, Universität Paderborn): Kombiniertes Hybrid-Monitoring und LAN-Analysing

Obwohl sich das Hybrid-Monitoring für die Detail- und Ursachenanalyse in heterogenen Umgebungen bewährt hat, gibt es keinen Anbietermarkt für derartige Monitoringsysteme, so dass sich die Entwicklung dieser Messwerkzeuge i.d.R. auf Hochschulen und Forschungseinrichtungen beschränkt. Ein speziell für derartige Messaufgaben entwickelter Hybrid-Monitor ist z.B. der Zählmonitor 4 (ZM4) (Universität Erlangen-Nürnberg), der auch bei Siemens eingesetzt wird. Der ZM4 wird als Hybrid-Monitor eingesetzt: Die Ereignisdefinition geschieht per Software, Messung und Aufzeichnung der Ereignisspur per Hardware. Der ZM4 kann beliebig viele, auch heterogene Prozessoren, die auch noch räumlich verteilt sein können, überwachen und gleichzeitig software- und hardwaregesteuerte Ereignisse aufzeichnen. Das Messdaten-Analysepaket SIMPLE bietet vielseitige Methoden zur Auswertung der ZM4-Ereignisspuren an, um aus den aufgezeichneten Spuren die interessierenden Leistungsgrößen, Konfliktsituationen, Ursachen für Engpässe u.a.m. ableiten zu können. ZM4 und SIMPLE sind universell einsetzbar, da sie nicht auf ein spezielles Rechensystem zugeschnitten sind.

Zur Analyse des Netzwerkes werden i.d.R. Netzwerkanalysatoren eingesetzt, wobei die Palette der auf dem Markt erhältlichen Test- und Messgeräte von kleinen portablen Geräten zur Überwachung der physikalischen Ebene bis hin zu komplexen Protokollanalysatoren mit der Möglichkeit der parallelen Überwachung mehrerer auch heterogener Netze reicht. Ein Beispiel für einen derartigen Netzwerk-Analysator mit einer modularen Architektur ist der DA30 (Wavetec Wandel Goltermann). Jeder der Bereiche Schnittstellenanpassung (Ethernet, FDDI etc.), Protokollanalyse und Bedienoberfläche wird dabei von je einem eigenständigen Prozessor gesteuert. Der Austausch der Daten zwischen dem Schnittstellenmodul und dem Protokollanalysemodul erfolgt über einen High-Speed-Bus, so dass eine Dekodierung der Protokolle auf allen sieben Schichten des OSI-Referenzmodells ohne Frameverlust in Echtzeit möglich ist.

Eine umfassende Performance-Analyse in Client/Server-Architekturen erfordert nicht nur die Überwachung des Ablaufgeschehens in den beteiligten Server- und Client-Systemen,

sondern auch die Aufzeichnung des Datenaufkommens auf dem physikalischen Übertragungsmedium sowie die Integration der Messergebnisse zu einer "globalen Systemsicht". Dazu wurde bei Siemens in Zusammenarbeit mit der Universität Paderborn ein Messadapter realisiert, der die Adaptierung des ZM4 an bis zu vier unterschiedliche Rechensysteme über eine Schnittstelle sowie die ereignisgesteuerte Steuerung des Netzwerk-Analysators ermöglicht. Um Betriebssystem- und Anwendungs-Software analysieren zu können, muss diese auf Hochsprachenebene um Messansatzstellen erweitert werden (Instrumentierung). Dies kann z.B. durch vordefinierte Makros geschehen, die bei der Abarbeitung ein Datum an die Drucker-Schnittstelle senden. Dieses Datum kann ein Steuerbefehl für den LAN-Analysator oder eine für den jeweiligen Programmbereich charakteristische Kodierung sein, die dann vom ZM4 mit einem Zeitstempel versehen abgespeichert wird. Dadurch wird erreicht, dass der LAN-Analyser nur so lange Daten protokolliert wie die zu analysierende Datenübertragung von dem Client-PC zu einem Server und wieder zurück dauert. So entstehen zwei verschiedenartige Messspuren, deren Einträge aber einander korrelieren. Nach Abschluss der Messung werden diese beiden Traces unter Berücksichtigung einer globalen Zeitbasis zu einer Mess-Spur zusammengeführt, die dann mit SIMPLE ausgewertet werden kann.

Mit dieser Messumgebung ist es möglich, als Quelltext vorliegende Programme zu instrumentieren, diese im Nano-Sekundenbereich zu vermessen und zusätzlich den resultierenden Netzwerk-Verkehr zeitlich zuordenbar aufzuzeichnen sowie die Mess-Spuren in einen Trace zu integrieren und weiterzuverarbeiten.

MyAMC (Siemens)

Das von Siemens entwickelte Werkzeug myAMC.LNI (auch R/3-Live Monitor genannt) ist ein Enterprise IT Management Werkzeug für SAP R/3-Systeme. Es können mit diesem Werkzeug verschiedene SAP R/3-Systeme von einer zentralen Stelle aus überwacht werden (Single Point of R/3 Control). Der R/3-Live Monitor besitzt die folgenden Eigenschaften, wenn er in dem Modus Basis Management betrieben wird:

- Übersichtliche Darstellung der Beziehungen, Verteilung und Aktivitäten von R/3-Datenbanken und ihrer zugehörigen R/3-Applikations-Instanzen
- Erkennung und Zuordnung von Performanceproblemen im R/3-System
- Anzeige von SAP R/3-Alerts (Alarmautomatik)

Eine Erweiterung stellt das Expert-Management dar, das zusätzlich zu den Basis-Management-Eigenschaften die Komponente „LNISCA“[3] beinhaltet. Die Erweiterung

[3] LNISCA = R/3-Live Monitor Service Quality, Capacity and Accounting

besteht in der Analyse der sogenannten R/3-Stat-Daten. Diese werden vom R/3-System standardmäßig für jede Transaktion in Form von Dialogschritten protokolliert.

Zur Darstellung der analysierten Dialogschritte bietet der R/3-Live Monitor verschiedene Sichten auf die Daten (vgl. Abbildung 04-3).

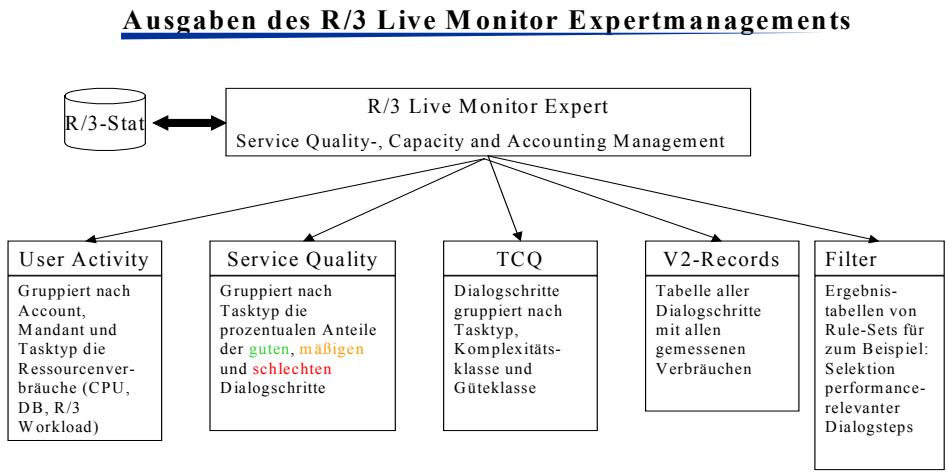

Ausgaben des R/3 Live Monitor Expertmanagements
R/3-Stat
R/3 Live Monitor Expert
Service Quality-, Capacity and Accounting Management
User Activity
Gruppiert nach Account, Mandant und Tasktyp die Ressourcenver-bräuche (CPU, DB, R/3 Workload)
Service Quality
Gruppiert nach Tasktyp die prozentualen Anteile der guten, mäßigen und schlechten Dialogschritte
TCQ
Dialogschritte gruppiert nach Tasktyp, Komplexitäts-klasse und Güteklasse
V2-Records
Tabelle aller Dialogschritte mit allen gemessenen Verbräuchen
Filter
Ergebnis-tabellen von Rule-Sets für zum Beispiel: Selektion performance-relevanter Dialogsteps

Abbildung 04-3: Ausgaben des R/3-Live Monitor Expertmanagements

WebMonitor (Universität Essen)

Unter der URL http://mapkit.informatik.uni-essen.de/oeffentl/WebMonitor.html steht ein Werkzeug zur Messung von Antwortzeiten beliebiger Web-Seiten zur Verfügung. Web-Monitor wurde an der Universität Essen entwickelt und führt die Antwortzeitmessung mit Hilfe von Javascript durch. Er kann mit Hilfe jedes gängigen Web-Browsers benutzt werden. Abbildung 04-4 veranschaulicht die Funktionsweise.

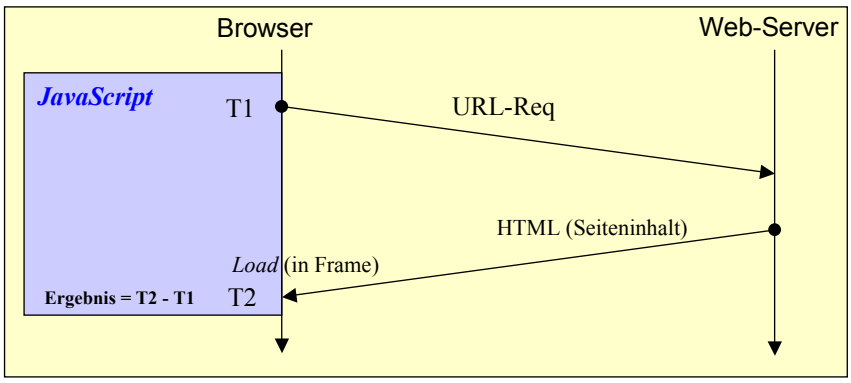


Abbildung 04-4: Funktionsweise des WebMonitors

Nach Eingabe einer beliebigen URL wird die zugehörige Seite geladen. Die Zeit vom Absetzen des Requests bis zum vollständigen Laden der Seite in den Browser wird gemessen und als Antwortzeit (Response Time) ausgegeben.

WebChecker (Universität Essen)

WebChecker ist ein speziell für Performance-Messungen konzipierter Web-Browser, der an der Universität Essen entwickelt wurde. Der Benutzer kann mit WebChecker in regelmäßigen Abständen (auch über längere Zeiträume hinweg) automatisiert die Performance von Web-Seiten überwachen. WebChecker erzeugt eine Trace-Datei, die die einzelnen Aktionen des Browsers wiedergibt. So wird das Laden verschiedener Objekttypen (images, applets etc.) mit den dazugehörigen Zeitangaben und bewegten Datenvolumina ausgewiesen. Die Trace-Datei kann von dem Auswertungswerkzeug Tracy interpretiert werden. Als Ergebnis können z.B. Tagesprofile für die Webserver-Antwortzeit erstellt werden.

Tracy (Universität Essen)

Tracy ist ein an der Universität Essen entwickelter Offline-Analysator für Ereignisspuren (Traces), d.h. ein Programm zur Offline-Analyse der Performance von Client/Server-Anwendungen. Es wird typischerweise im Rahmen eines Transaktionsmonitorings eingesetzt, wobei entsprechend instrumentierte Anwendungen (wie z.B. WebChecker) während des laufenden Systembetriebs ihre Performance aufzeichnen.

Tracy ist nicht auf einen bestimmten Anwendungstyp beschränkt. Die Voraussetzung ist eine instrumentierte Anwendung, die Ereignis-Traces in einem für Tracy lesbaren Format erzeugt. Die Instrumentierung der Anwendung erfordert einen relativ geringen Eingriff in den Source-Code, da sie keinerlei Berechnungen von Performance-Werten beinhaltet, son-

dern lediglich die Protokollierung einzelner Ereignisse mit dazugehörigen Zeitstempeln und ggf. zusätzlichen Informationen wie z.B. transferiertes Datenvolumen gewährleisten soll.

Für die Analyse von WebChecker-Traces existiert eine spezielle Version WebTracy, die eine grafische Benutzerschnittstelle besitzt und eine effiziente Auswertung auch komplexer Web-Seiten liefert.

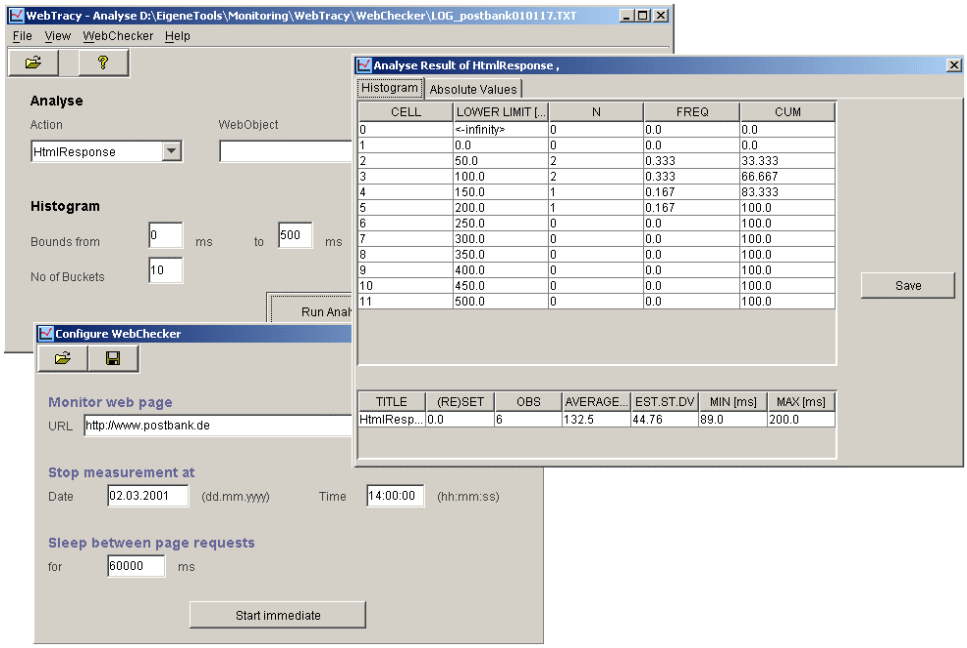

WebTracy - Analyse D:\EigeneTools\Monitoring\WebTracy\WebChecker\LOG_postbank010117.TXT
File View WebChecker Help
Analyse
Action
WebObject
HtmlResponse
Histogram
Bounds from 0 ms to 500 ms
No of Buckets 10
Run Anal
Analyse Result of HtmlResponse ,
Histogram
Absolute Values
CELL | LOWER LIMIT [... | N | FREQ | CUM
0 | <-infinity> | 0 | 0.0 | 0.0
1 | 0.0 | 0 | 0.0 | 0.0
2 | 50.0 | 2 | 0.333 | 33.333
3 | 100.0 | 2 | 0.333 | 66.667
4 | 150.0 | 1 | 0.167 | 83.333
5 | 200.0 | 1 | 0.167 | 100.0
6 | 250.0 | 0 | 0.0 | 100.0
7 | 300.0 | 0 | 0.0 | 100.0
8 | 350.0 | 0 | 0.0 | 100.0
9 | 400.0 | 0 | 0.0 | 100.0
10 | 450.0 | 0 | 0.0 | 100.0
11 | 500.0 | 0 | 0.0 | 100.0
Save
TITLE | (RE)SET | OBS | AVERAGE... | EST.ST.DV | MIN [ms] | MAX [ms]
HtmlResp... | 0.0 | 6 | 132.5 | 44.76 | 89.0 | 200.0
Configure WebChecker
Monitor web page
URL http://www.postbank.de
Stop measurement at
Date 02.03.2001 (dd.mm.yyyy) Time 14:00:00 (hh:mm:ss)
Sleep between page requests
for 60000 ms
Start immediate

Abbildung 04-5: GUI von WebTracy

Snoop-GUI (Materna)

SNOOP ist ein unter UNIX zur Verfügung stehendes Werkzeug für die Analyse und die Auswertung von TCP/IP-Netzverkehr. Mittels einer im Rahmen von MAPKIT entwickelten, auf Java basierenden Oberfläche besteht die Möglichkeit, die umfassende SNOOP-Syntax übersichtlich anzuwenden. Die Bedienung von SNOOP wird so vereinfacht. Die Verkehrsstatistiken können nach bestimmten Kriterien gefiltert und übersichtlich dargestellt werden.

Dabei wird der Kommunikation zwischen Rechnersystemen besondere Beachtung geschenkt. Der Verbindungsaufbau, die Durchführung des Datentransfers und die Beendi-

gung der Verbindung wird verdeutlicht. Der Datenverkehr wird durch SNOOP aufgezeichnet, in ein Logfile geschrieben und dann durch die grafische Benutzerschnittstelle anschaulich aufbereitet. Die Integration mit Tracy ermöglicht die bereits beschriebene Offline-Analyse der SNOOP-Logfiles.

Monitoring- und Diagnosetool GlancePlus (HP)

GlancePlus ist ein System Performance-Monitor und Diagnose-Werkzeug. Das Programm gehört zu den HP OpenView Ressourcen- und Performance-Produkten. Es ist ein Online Performance-Monitor, mit dem eventuell vorhandene Systemengpässe angezeigt und die Verursacher (Prozesse) ermittelt werden können.

GlancePlus ist für die Plattenformen HP-UX, Solaris, AIX und Reliant UNIX verfügbar. Während der Erprobung wurde es nur auf HP-Systemen betrieben. Ob alle Funktionen und Messwerte auf allen Plattformen zur Verfügung stehen, kann deshalb nicht bewertet werden.

Mit GlancePlus steht ein sehr gutes Tool für Performance-Analysen zur Verfügung. Mit den über 1000 Messpunkten ist eine sehr detaillierte Diagnose bei System-Performance-Problemen möglich. Im Vergleich zu anderen Standardmesstools im UNIX-Umfeld ist die prozessspezifische Zuordnung zahlreicher Messgrößen zu unterstreichen. Nachteilig ist, dass standardmäßig in der LAN-Statistik zusätzlich zur Paketanzahl keine Messwerte für das Nachrichtenvolumen (Anzahl übertragener Bytes) vorhanden sind. Hervorzuheben ist insbesondere die sehr nützliche Online-Erklärung zu den einzelnen Messwerten. Gute Performance-Kenntnisse vorausgesetzt, ist das Tool relativ leicht zu erlernen und zu bedienen.

Überwachung von Transaktionen/Geschäftsprozessen

Es besteht die Möglichkeit, sich Messdaten wie

- mittlere Transaktionszeit und Verteilung der Transaktionszeit,
- Anzahl Transaktionen und
- Einhaltung von Service Levels (z.B. Transaktionszeit kleiner 2 Sekunden) vom MeasureWare Agent über das Interface ARM (Application Responce Measurement) anzeigen zu lassen.

Die Applikationen müssen dazu mit dem Transaktion-Tracker vorbereitet sein. Der Transaktion-Tracker ist Bestandteil des Applikation Programm Interfaces (API), mit dem der Programmierer Transaktionen innerhalb von Applikationen definieren kann.

Abgrenzung zu den Standardmesstools in UNIX:

Glance benutzt spezielle Trace-Funktionen im HP-UX Kernel und setzt diese in Messwerte über den midaemon-Prozess um. Somit können wesentlich genauere und zusätzliche Messdaten gegenüber den Standardtools wie sar, vmstat und iostat zur Verfügung gestellt werden. Diese „kmem"-Standardtools ermitteln ihre Daten aus Zählern im Kernel über ein Sampling-Verfahren. Mit Glance stehen alle Messwerte der einzelnen Tools über eine einheitliche Schnittstelle zur Verfügung.

04.02 Werkzeuge zur Planung und Prognose

(a) Kategorisierung von Verfahren bzgl. Aufwand und Genauigkeit

Techniken und Werkzeuge zur Planung und Prognose differieren in drei wesentlichen Aspekten. Zum einen ist der Aufwand zur Erstellung der Prognosebasis (Modell) zu berücksichtigen. Damit eng verbunden sind die Genauigkeit der Prognoseergebnisse und die Kosten zur Lösung der Modelle, siehe Abbildung 04-6.

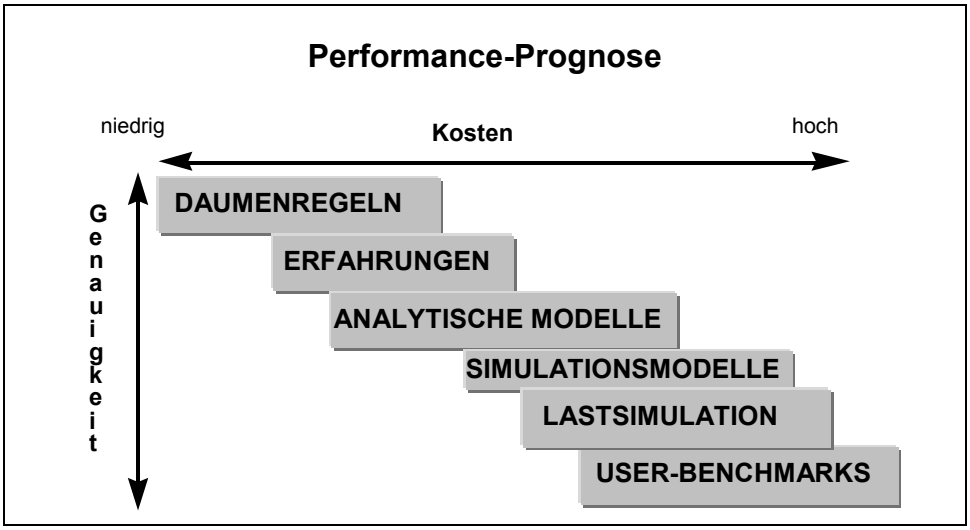


Abbildung 04-6: Kategorisierung von Prognoseverfahren

Daumenregeln

Hierbei handelt es sich um allgemeine Regeln zur Vorhersage des Performance-Verhaltens im Großen und Ganzen. Dabei werden oft nur die zentralen Komponenten und diese auch nur separat betrachtet, um exzessive Verzögerungen zu vermeiden. Daumenregeln helfen, Grenzen abzustecken und können daher als Richtlinien aufgefasst werden. Sie erfordern wenig Aufwand und damit geringe Kosten, liefern aber entsprechend ungenaue Ergebnisse. Vielfach werden Erfahrungen und Angaben von Herstellern und Betreibern herangezogen, um Daumenregeln zu formulieren.

Trendanalyse

Diese Technik gewinnt Performance-Vorhersagen basierend auf der Beobachtung der Historie. Dazu werden Langzeitbeobachtungen des Systems mit Hilfe von Monitoring-Werkzeugen vorgenommen und die gewonnenen Daten entsprechend verdichtet. Häufig werden lineare Extrapolierungsansätze verwendet. Dieser Ansatz, den Zusammenhang zwischen Last und Performance-Maßen durch eine lineare Gleichung zu beschreiben, mag für schwach ausgelastete Systeme richtig sein. Allerdings berücksichtigt dieser Ansatz nicht den rasanten Anstieg von Leistungsmaßen in der Nähe der Sättigungspunkte. Darüber hinaus erfasst die Trendanalyse in der Regel nicht das Zusammenwirken unterschiedlicher Arbeitslasten. Der Aufwand für eine Trendanalyse ist höher als der Aufwand für Daumenregeln, da für diesen Prognoseansatz Messdaten erhoben werden müssen. Auf Grund der oben beschriebenen Probleme sind die Ergebnisse vor allem für hoch ausgelastete Ergebnisse fraglich.

Lastsimulation, User Benchmarking, Analytische und Simulative Verfahren

Diese Methoden werden im Folgenden detailliert behandelt.

(b) Lastsimulation und User Benchmarking

In vernetzten heterogenen Systemen werden heute häufig Lastsimulation und User Benchmarking zur Absicherung der funktionalen Korrektheit und einer zufriedenstellenden Systemleistung angewendet.

Lastsimulation beinhaltet die Nachbildung realer Applikationen und Datenbestände sowie des Benutzerverhaltens. Im Realbetrieb werden die Lastdaten erhoben, mit denen die Clients die Netzwerke und die Server-Systeme belasten. Darüber hinaus werden die Lastdaten erhoben, mit denen die Serversysteme die Netzwerke belasten. Die Synthetisierung erfolgt mittels Lastgeneratoren, die auf Basis dieser Lastdaten die entsprechenden Netz- und Serverlasten produzieren. Der Ablauf erfolgt i.d.R. in einer der Realität nachgebildeten Testumgebung. Messwerkzeuge analysieren das Systemverhalten hinsichtlich Verfügbar-

keit, Zuverlässigkeit und Performance und ermöglichen so die Bewertung und Dimensionierung des Gesamtsystems. Der Aufwand ist vor allem für die Auswertung der erhobenen Last- und Leistungsdaten und deren Umsetzung in die synthetischen Lastprofile sowie für die Validierung und Kalibrierung dieser Profile sehr hoch. Deshalb wird die Lastsimulation insbesondere zur Analyse des Systemverhaltens von vernetzten IT-Strukturen eingesetzt und zwar dann, wenn einerseits sensible Daten vorliegen oder die realen Applikationen nicht eingesetzt werden können bzw. sollen und andererseits zur Generierung von Hoch- und Extremlastfällen, um nicht nur die Performance, sondern auch die Verfügbarkeit und Zuverlässigkeit der System- und Netzwerkkomponenten zu analysieren und zu bewerten.

Benchmarks umfassen eine Menge von "repräsentativen" Programmen oder Anwendungen und werden unterschieden nach "Standard-Benchmarks" und "Kunden-Benchmarks". Standard-Benchmarks sind offiziell standardisiert (u.a. SPEC, TPC) und dienen insbesondere zum Performance-Vergleich unterschiedlicher IT-Systeme. Beim Upgrading werden die Benchmark-Ergebnisse häufig zur Prognose der Skalierbarkeit und zur Dimensionierung herangezogen. Ihre Aussagekraft liegt zwischen den o.g. Verfahren Daumenregel und Trendanalyse.

Kunden-Benchmarks bauen auf realen Applikationen und Datenbeständen auf und stellen i.d.R. ein repräsentatives Abbild der realen Last dar. Sie werden sehr häufig zur Absicherung einer Performance-gerechten Dimensionierung von Serversystemen verwendet. Zur Nachbildung der realen Benutzer und deren Verhaltens werden häufig Treibersysteme eingesetzt, die in die Lage sind, auf Basis von vorher am Client-PC aufgezeichneten, repräsentativen Benutzerdialogen, eine Vielzahl von Benutzern synthetisch nachzubilden. Die so erzeugte Last lässt sich vervielfältigen. Abhängig von der real darunter befindlichen Systemumgebung ergibt sich das an der Realität nachgebildete Lastverhalten.

Zur Dimensionierung des Netzwerkes und damit des Gesamtsystems ist u.a. wegen der Sequentialisierungseffekte der Einsatz eines Treibersystems nur bedingt geeignet. Last- und Performance-Tests in vernetzten IT-Umgebungen werden deshalb zweckdienlich mit einer großen Anzahl (>> 100) von Clientsystemen durchgeführt, um so möglichst realitätsnah und vielseitig Lasten erzeugen zu können. Insbesondere ist die Einbeziehung des Netzwerkes bei solchen Tests ein nicht zu vernachlässigender Aspekt, dem nur mit der Beteiligung von sehr vielen Netzknoten (Clients) Rechnung getragen wird. Erst die Möglichkeit, an vielen Clients Lasten zu erzeugen, die alle nahezu gleichzeitig oder auch nur sporadisch auf das Netzwerk und die Server-Systeme zugreifen, können Funktionalität und Performance des vorhandenen Netzes und somit das Verhalten des Gesamtsystems entsprechend bewertet werden. Die Durchführung derartiger Last- und Performance-Tests macht es erforderlich, dass Mechanismen zur definierten Lastgenerierung sowie zur Ansteuerung und Überwachung der PCs verfügbar sind. Diesbezüglich bestehen spezifische Anforderungen an die Testautomatisierung hinsichtlich Lasterzeugung, Reproduzierbarkeit und Variierbarkeit

sowie Konsistenz und Überprüfbarkeit der Ergebnisse. Bei Beteiligung vieler Clients empfiehlt sich eine Messautomatisierung in Form einer zentralen Steuerungseinheit zur Synchronisation der Clients, Steuerung der gesamten Umgebung, um die Kontinuität in der Erzeugung einer gleichförmigen Last zu gewährleisten.

Universeller Lastgenerator SyLaGen (Siemens)

Die Komplexität vernetzter Systeme nimmt ständig zu. Es treten hier immer wiederkehrende Fragestellungen der IT-Betreiber und -Anwender auf wie:

- Welche Auswirkungen haben Hardware- oder Software-Umstellungen auf die Funktionalität, die Verfügbarkeit und die Performance des Gesamtsystems?
- Reicht die Hardware-Kapazität der System-Ressourcen und Netzbandbreiten für die nächste Ausbaustufe der Anwendungsumgebung aus?

Lösungskonzept SyLaGen:

SyLaGen (Synthetischer Lastgenerator) bietet eine voll automatisierte Steuerungs- und Lastgenerierungsumgebung für die oben genannten Anforderungen. Das Werkzeug erlaubt die Spezifizierung heterogener Lastprofile über eine einheitliche Schnittstelle für alle Anwendungsfälle. Die Erzeugung verschieden gearteter Anwenderlasten, wie z.B. File-I/O, TCP/IP-Kommunikation, HTTP-Kommunikation oder auch einer aufgezeichneten realen Kundenanwendung wird über entsprechende Module realisiert.

SyLaGen besteht aus einer beliebigen Anzahl von Client-Agenten als Lastgeneratoren, einem oder mehreren Server-Agenten und einem Master. Die Lastgeneratoren laufen im Rahmen einer bereits installierten und konfigurierten Systemumgebung auf den Client-Systemen ab und bringen so eine definierte, reproduzierbare Last auf das Netz und die involvierten Server-Systeme. Dabei können auch Konfigurationen simuliert werden, deren Hardware nicht in vollem Umfang zur Verfügung steht. Alle Testläufe bzw. Messungen werden zentral vom Master gestartet und anschließend von diesem bedienerlos gesteuert, synchronisiert, überwacht und ausgewertet. Ein Server-Agent sammelt Server-spezifische Performance-Daten. Abbildung 04-7 verdeutlicht die Architektur von SyLaGen.

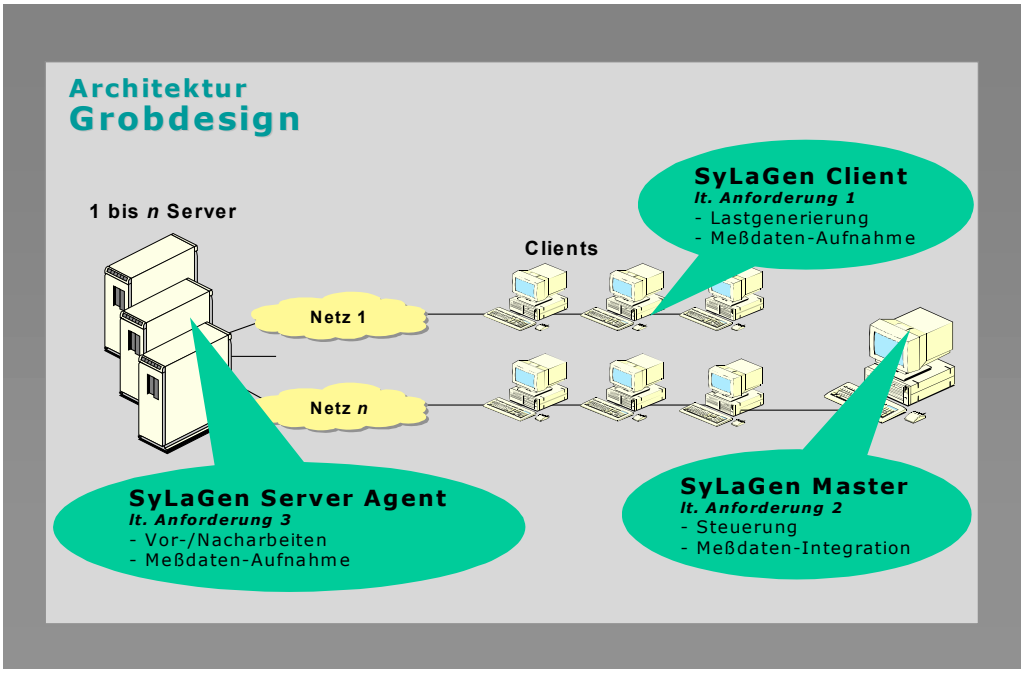


Abbildung 04-7: Architektur von SyLaGen

Mit SyLaGen können Tests bzw. Messungen sowohl mit fest vorgegebener Anzahl von Clients (Lastgeneratoren) als auch - in Form eines Benchmarklaufs - mit einer automatisch variierenden Anzahl von Clients durchgeführt werden. Bei einem solchen Benchmarklauf wird automatisch in der vorliegenden Systemumgebung mit dem kundenspezifisch festgelegten Lastprofil und vorgegebenem Transaktionsdurchsatz die maximale Anzahl performant betreibbarer Clients ermittelt. Beurteilungskriterium hierfür ist die Einhaltung einer vorgegebenen Bearbeitungszeit.

SyLaGen liefert umfangreiche Ergebnisdaten in Form von

- Konfigurationsreports aller beteiligten Systeme,
- Reports über die Ressourcen-Verbräuche aller Server-Systeme sowie
- Statistiken über die erzeugten Lastprofile und die Bearbeitungszeiten der aktiven Clients.

SyLaGen ermöglicht es, ein gegenwärtiges oder geplantes Belastungsprofil in Form einer Lastsimulation nachzubilden. So wird die Anwendung im Gesamtsystem hinsichtlich des zu erwartenden Systemverhaltens getestet. Auf der Basis des spezifizierten Belastungspro-

fils können unterschiedliche Hardware- und Software-Umgebungen hinsichtlich Funktionalitäts-, Verfügbarkeits- und Performanceveränderungen im Systemverhalten untersucht und verglichen werden. Darüber hinaus zeigen Stresstests das Systemverhalten während maximaler Belastung auf. Ist die konkrete Nachbildung des kundenspezifischen Belastungsprofils zu aufwändig, können vordefinierte Lastprofile für unterschiedliche Applikationen verwendet werden.

PerformanceStudio (Rational)

„PerformanceStudio“ ist ein Tool der Firma Rational zur Durchführung von Lasttests von e-Business-, Client-Server- und Enterprise Resource Planning-Applikationen. Der Aufbau und die Funktionalität der Durchführung von User-Benchmarks ist vergleichbar mit den Produkten „Loadrunner“ der Firma Mercury und „QALoad“ von Compuware.

PerformanceStudio ist für Lasttests und Benchmarks im speziellen Bereich von verteilten Client-Server-Applikationen gut einsetzbar. Der Fokus liegt darauf, große Anzahlen konkurrierender Benutzer reproduzierbar nachzubilden, ohne jedoch das physikalische Equipment in Form von Clients vorhalten zu müssen.

Wie schon das Vorgängerprodukt basiert PerformanceStudio auf der Agenten-Technik mit UNIX oder NT Agenten. Gesteuert werden diese durch eine NT Master Station. Im einfachsten Fall sind Master Station und Agent auf einer Maschine.

Mittels einer Capture-Facility werden bei der Aufzeichnung typischer Benutzerinteraktionen entweder die höheren Netzwerk-Protokolle wie ORACLE-SQL*Net, TUXEDO und HTTP, das diesen Protokollen zu Grunde liegende Socket-Protokoll oder GUI-Befehle aufgezeichnet und in Skripten abgelegt.

Die Skripte sind, je nach Wahl der Aufzeichnung (Netzwerkprotokoll oder GUI), ähnlich einem C- oder VisualBasic-Programm. Diese können dann weiter editiert und mit Kontroll-Strukturen sowie Variablen versehen werden. Die übersetzten Programme sind auf dem Agenten-Rechner ablauffähig und emulieren dann einen oder mehrere Clients. Während des Ablaufs der Programme werden Statistiken über definierte Zeitstrecken und die automatisch generierten Ablauftests gesammelt. Damit werden Antwortzeit- und Test-Statistiken erzeugt, die auf der Master Station mit einer Referenz zum Skript in einer Datenbank abgelegt werden.

Folgendes Bild zeigt den schematisierten Ablauf beim User-Benchmark:

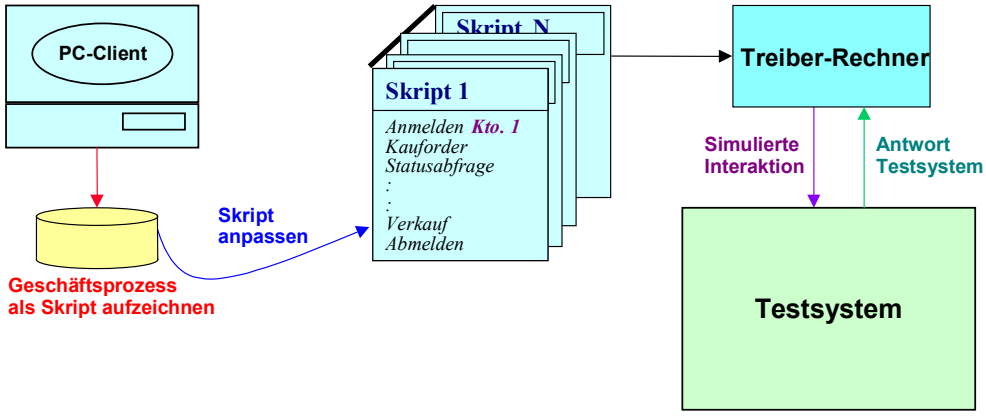


Abbildung 04-8: Ablauf eines User-Benchmark

(c) Modellierungswerkzeuge

Bemerkungen zu analytischen und simulativen Verfahren

Analytische Modellierung beschreibt die Performance von Rechensystemen durch eine Reihe von Gleichungen, für die sich exakte oder approximative Lösungen angeben lassen. Aufgrund der Lösungsstrategie ist der Aufwand zur Lösung der Modelle gering. Oft benötigt man zur Modelllösung nur wenige Sekunden. Der Aufwand zur Erstellung der Modelle kann sehr groß sein, da in der Regel eine Vielzahl von Abstraktionen, Vereinfachungen und Annahmen nötig ist, um das Performance-Verhalten komplexer Systeme auf einige wenige Parameter zu reduzieren. Die Genauigkeit der Modelle hängt sehr stark davon ab, wie genau sich die Realität durch die Elemente der Modellklasse beschreiben lassen, die zur Modellierung herangezogen wird. Die vollständige Festlegung der Eingangsgrößen beschränkt sich in der Regel auf wenige Parameter, die messtechnisch erfasst werden können. Für den Ansatz einer analytischen Modellierung existieren eine Reihe universitärer und kommerzieller Werkzeuge, die für gängige Netzkomponenten und Link-Layer-Technologien fertig parametrisierte Bausteine anbieten.

Simulative Modelle ermöglichen es, genaue Performance-Ergebnisse zu erzielen, da sie, falls gewünscht, alle Details des realen Systems nachbilden können. Allerdings führt das nicht nur zu einem erhöhten Aufwand zur Lösung des Modells, da somit die Anzahl der zu simulierenden Ereignisse ansteigt, sondern auch zu einem erhöhten Aufwand bei der Gewinnung von Eingangswerten. Weiterhin spiegeln möglicherweise exakt nachgebildete Simulationsszenarien eine Modellgenauigkeit wider, die sich aufgrund fehlender oder geschätzter Eingangswerte als falsch herausstellt. Zur Ermittlung von Leistungsmaßen mit

einer sehr hohen Vertrauenswürdigkeit sind ebenso extrem lange Simulationsläufe notwendig wie zur Ermittlung von Ergebnissen für extrem seltene Ereignisse. Daher sollten Simulationen hauptsächlich zur Analyse und Bewertung von wichtigen Details eingesetzt werden.

BEST/1 (BMC)

BEST/1 ist ein Werkzeug, das sich im Mainframe-Bereich seit über 20 Jahren erfolgreich behauptet und seitdem kontinuierlich weiter entwickelt hat. Der Urheber von BEST/1, Firmengründer und Chef-Theoretiker Jeff Buzen, verdankt seinen Geschäftserfolg der Fokussierung auf Großkunden mit geschäftskritischen interaktiven Diensten; in der Historie waren dies v.a. Fluggesellschaften und Banken. Außerdem wurden schon früh problemangepasste Bedienoberflächen und die enge Verzahnung von SNMP-basierten Mess- und Monitoringmethoden mit Modellierungstechniken realisiert. Messdaten werden an die Auswerte- und Visualisierungskomponenten BEST/1-Monitor und Visualizer übermittelt und zur Konstruktion und Auswertung von Prognosemodellen verwendet. Die Auswirkungen von Konfigurations- und Laständerungen können dann per Modellanalyse durch die Komponente BEST/1-Predict berechnet werden.

Neben der klassischen Mainframe-orientierten Variante des Werkzeugs existiert inzwischen auch BEST/1 for Distributed Systems. Die Übernahme von BEST/1 durch BMC Anfang 1998 zeigt den Trend des Zusammenwachsens von Modellierungs- und Prognosewerkzeugen.

EcoPredictor (Compuware)

EcoPredictor (früher COMNET-Predictor von CACI) ist ein Werkzeug, welches auf die Analyse von Netzwerken spezialisiert ist. Predictor-Modelle sind aufgebaut wie die Pläne der System- und Netzadministration, d.h. es gibt Icons für Router, Switches, LANs, Subnetze, Hosts, Server und Clientgruppen. Die Verkehrslast wird durch sog. Paketflüsse spezifiziert, welche sich gemäß der gängigen Routingstrategien (z.B. kürzeste Wege) durch das Netz bewegen. Die besondere Herausforderung für den Anwender besteht darin, sinnvolle Parametrisierungen von Modellbausteinen und Lastbeschreibungen zu ermitteln. Mit dem Partnerwerkzeug COMNET Baseliner können u.a. automatisch erfasste Topologiedaten als Grundlage für Modellentwicklungen genutzt werden. Die Stärken des Predictors sind seine Schnelligkeit - mehrere Prognoseperioden werden sekundenschnell berechnet - und die detaillierte und zielgerichtete Auswertemöglichkeit. Es können z. B. Auslastungen und Durchsätze für alle Komponenten getrennt nach Szenarien ermittelt werden.

COMNET III (Compuware)

Differenziertere, aber wesentlich aufwändigere Modellexperimente können mit dem Simulator COMNET-III durchgeführt werden. Der Simulator ermöglicht eine wesentlich detailliertere Beschreibung von System- und Lasteigenschaften, weshalb ein höherer Aufwand für die Erstellung, Auswertung und Pflege der Modelle berücksichtigt werden muss. Für den COMNET-III-Simulator werden eine Reihe von spezialisierten Add-On-Packages, so z.B. für Client/Server-Systeme, für ATM-Netze und für Satelliten-Kommunikation angeboten.

SES/strategizer (HyPerformix)

Dem COMNET-III-Simulator vergleichbar ist der SES/strategizer, welcher von vornherein als Client/Server-Werkzeug konzipiert ist. Die Parameter aller Bausteine, wie z.B. Angaben zur Leistungsfähigkeit von Prozessoren oder Netzkomponenten sind Bestandteil der Distribution und müssen nicht als separates Modul hinzugekauft werden. In jedem Falle sind diese Bausteine aber permanent zu pflegen, zu validieren und bzgl. des Einsatzbereiches anzupassen. Nur dann sind vertrauenswürdige Prognoseergebnisse zu erzielen. Die Problematik der Modellvalidation ist eine dem modellgestützten Ansatz inhärente Eigenschaft, mit der alle Werkzeuge gleichermaßen konfrontiert sind.

OPNET (MIL3)

Die Werkzeuge OPNET Planner und OPNET Modeler von MIL3 bieten ähnliche Funktionalitäten wie die bereits genannten Modellierungswerkzeuge. Der Planner ist vorwiegend für die schnelle Simulation von Netzwerken aus vorgefertigten Bausteinen konzipiert, während der Modeler zusätzlich die Definition von modifizierten und benutzerspezifischen Bausteinen zulässt. Insbesondere ist auch die hierarchische Spezifikation von großen, komplexen Systemen bis hin zur Beschreibung von Protokollsoftware explizit möglich. In umfangreichen Bibliotheken werden u.a. Modellbausteine angeboten für Protokolle, Netzwerkkomponenten und Verkehrsquellen. Beispielweise gibt es in Form des OPNET Modeler/Radio eine Version, welche Bausteine speziell für die Analyse von Mobilnetzen mit Radio- und Satellitenverbindungen enthält. Durch eine strategische Allianz von MIL3 und HP versprechen die OPNET Tools besonders gut mit HP OpenView und HP NetMetrix zu kooperieren.

VITO (Universität Essen, Siemens)

VITO ist ein an der Universität Essen im Rahmen von MAPKIT entwickeltes Werkzeug zur Planung und Prognose von IT-Systemen. Es folgt dem Modellparadigma der Warteschlangensysteme. Das Modell wird durch eine Anzahl von Stationen und Ketten beschrieben. Die Stationen beschreiben die Systemkomponenten, die Ketten die Systemlast.

Abbildung 04-9 zeigt die grafische Darstellung der Stationen. Die gleiche Abbildung zeigt das sog. Mapping, welches die Bedienanforderungen der Lastkette, hier der Kette „Dialog“, an den Stationen beschreibt.

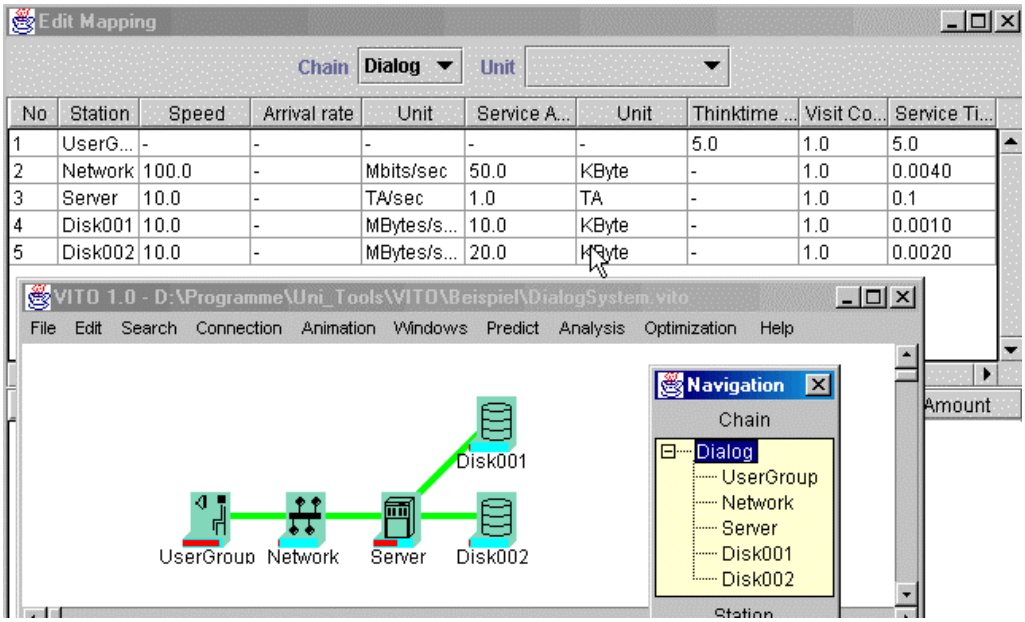


Abbildung 04-9: VITO Modell

Die VITO-Modelle werden mit Hilfe eines approximativen analytischen Verfahrens (sogenannter Baard-Schweitzer-Algorithmus) gelöst. VITO besitzt eine grafische Benutzeroberfläche und stellt Bausteine für verschiedene typische IT-Systemkomponenten zur Verfügung.

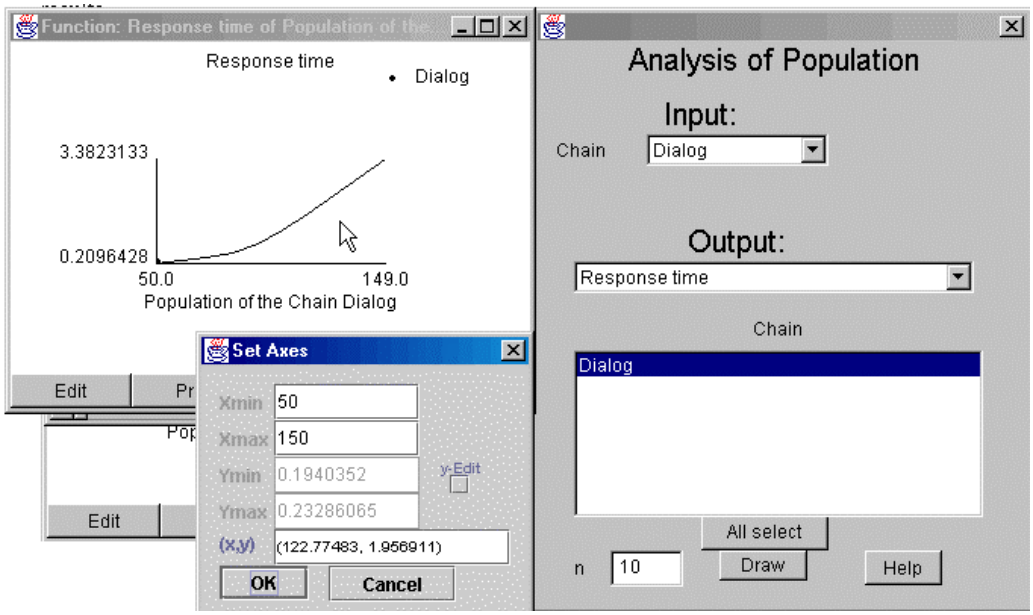


Abbildung 04-10: VITO Modellierungsergebnisse

VITO schließt die Lücke zwischen abstrakten reinen Warteschlangenwerkzeugen und kommerziellen, spezialisierten Modellierungswerkzeugen/-Frameworks, indem eine Modellierung auf verschiedenen Abstraktionsebenen zugelassen wird, die Modelle aber in den Kontext der IT-Systeme eingebunden sind. VITO besitzt eine Eingabeschnittstelle zum Messwerkzeug PERMEX von Siemens und eine Ausgabeschnittstelle zu Excel. Eine der speziellen Funktionen von VITO ist die Möglichkeit zum interaktiven Experimentieren im Sinne von „What-if-Analysen“. Abbildung 04-10 zeigt rechts das Dialogfenster, in dem festgelegt ist, dass die Kette „Dialog“ bzgl. ihres Antwortzeitverhaltens (Output Parameter: Response time) untersucht wird. Der variierte Parameter ist die Population, die hier im Bereich von 50 bis 150 Aufträgen liegt (Auftrag = aktiver Dialog-Benutzer). Auf Grundlage von 10 Analyseläufen (Rechenzeit ca. 1 Sekunde) wird das Diagramm links oben erstellt, welche den Anstieg der Antwortzeit als Funktion der Population zeigt.

WLPSizer (Universität Essen, Siemens)

Der WLPSizer („Workload Profile Sizer“) ist ein Werkzeug zur Kapazitätsplanung von SAP R/3-Client/Server-Systemen. Eine der Hauptfragen, die sich im R/3-Bereich stellt und die mit Hilfe des WLPSizers beantwortet werden kann, ist: „Können die Arbeitslasten mit vom Kunden geforderter Dienstgüte durch eine spezifizierte R/3-Konfiguration bewältigt

werden, bzw. wie muss diese Konfiguration zur Bewältigung der Arbeitslast modifiziert werden?“

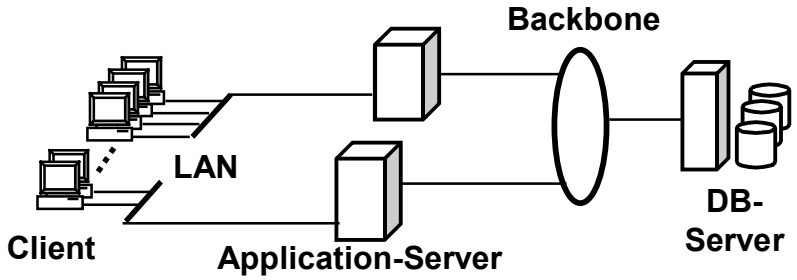


Abbildung 04-11: Mehrstufige R/3-Konfiguration

Der WLPSizer ist ein „Schwester-Werkzeug“ von VITO, da beide denselben approximativen Modelllösungsalgorithmus verwenden. Die grafische Benutzeroberfläche ermöglicht die R/3-spezifische Systemmodellierung. Der WLPSizer verfügt über zahlreiche R/3-typische Bausteinbibliotheken. Für die errechneten Leistungsmaße existieren vier verschiedene Sichten: Server-View, I/O-View, Workload-View, Utilization-View. Zum Beispiel werden in der Server-View zu den verschiedenen Tasktypen Prognosewerte erstellt für:

- Systemdurchsätze TA/h
- Antwortzeiten der Tasks (Workprozesse)
- Verweilzeiten am Application-Server
- Auslastungen und Verzögerungen bzgl. der Netze
- Mittlere Verweilzeit am DB-Server

Der WPLSizer besitzt darüber hinaus eine Schnittstelle zum Einlesen von Daten des R/3-Messwerkzeugs myACM.LNI von Siemens und eine Ausgabeschnittstelle zu Excel.

04.03 Zusammenspiel von Mess- und Prognose-Werkzeugen

(a) Ausgangssituation und Zielsetzung

Bei Messung, Modellierung und Prognose fallen im Wesentlichen die folgenden Arbeitsschritte an:

- Erhebung von Messdaten und deren Auswertung

@ Übernahme relevanter Daten in Planungs- und Prognosewerkzeuge

@ Durchführung von Planung und Prognose

Die Erfahrung zeigt, dass gerade die manuelle Übernahme der notwendigen Eingangsdaten in Prognosewerkzeuge ein sehr zeit- und arbeitsintensiver Prozess ist.

Eine wesentliche Optimierung der oben aufgezeigten Arbeitsschritte erreicht man durch eine automatisierte Erfassung und Übernahme der erforderlichen Eingangsdaten. Hierbei sind Last, Konfiguration und Topologie des zu untersuchenden Systems in entsprechende Modellbausteine und -Parameter des entsprechenden Analysewerkzeuges umzusetzen. Diesen Prozess bezeichnet man auch als automatisiertes Baselining**.** Auf diese Weise wird folgender Nutzen erreicht.

@ Deutliche Verbesserung der Zeit- und Kosteneffizienz im gesamten Planungsprozess

@ Konzentration auf den Kern der eigentlichen Planungsaufgabe

(b) Spiegelung der Anforderungen an der Funktionalität marktgängiger Prognosewerkzeuge

Das Prognosetool COPE (Siemens)

COPE ist ein zur Leistungsprognose für BS2000-Rechnerkonfigurationen zugeschnittenes werkzeuggestütztes Verfahren. In den letzten 15 Jahren wurden mit COPE mehr als 3000 Prognose- und Dimensionierungs-Projekte erfolgreich durchgeführt. Die breite Akzeptanz, welche dieses Werkzeug beim praktischen Einsatz erfahren hat, ist im Wesentlichen auf folgende Funktionsmerkmale zurückzuführen:

@ Liegt eine reale BS2000-Anwendung als Referenz für ein Modell vor, so ist diese mit dem Softwaremonitor COSMOS zu vermessen. COPE verfügt über ein eigenes Auswerteprogramm und erstellt die Basisdaten für ein Modell des IST-Zustandes automatisch. Dieser Weg ist schnell und genau. Der Modellierungsaufwand reduziert sich hier gegenüber Tools, die eine derartige automatisierte Übernahme der Inputdaten nicht unterstützen bei vergleichbaren Aufgabenstellungen auf etwa 1/7.

@ Insbesondere die Abbildung von Geschäftsprozessen auf IT-Prozesse und die verursachergerechte Zuordnung von den für einzelne Geschäftsprozesse feststellbaren Ressourcen-Verbräuchen zu den involvierten Hardwarekomponenten wird hier weitgehend automatisiert.

@ Ein wesentlicher Vorteil ist, dass die gesamten relevanten Informationen zur Hardware (Host mit Plattenperipherie und die direkt angeschlossenen Netzkomponenten), zur

Systemsoftware und die aktuell pro Geschäftsprozess anfallenden Ressourcen-Verbräuche zentral an einer einzigen Stelle gesammelt werden können.

@ Interne Strategien und Abläufe des BS2000 sind in COPE integriert. Das Verfahren greift auf eine umfangreiche Datenbasis mit aktuellen Produktkennwerten, Systemstandards, Konfigurationsregeln und Performanceempfehlungen zu und unterstützt den Anwender Online bei der Durchführung seiner Prognoseaufgaben. Die technische Umsetzbarkeit der im Modell nachgebildeten BS2000-Konfiguration in eine reale Konfiguration ist somit weitestgehend sicher gestellt.

Unten stehende Abbildung 04-12 skizziert den Ablauf Messen, Modellieren und Planen und zeigt die wesentlichen Elemente, die im Falle von COPE automatisiert von der Messung in ein Modell übernommen werden.

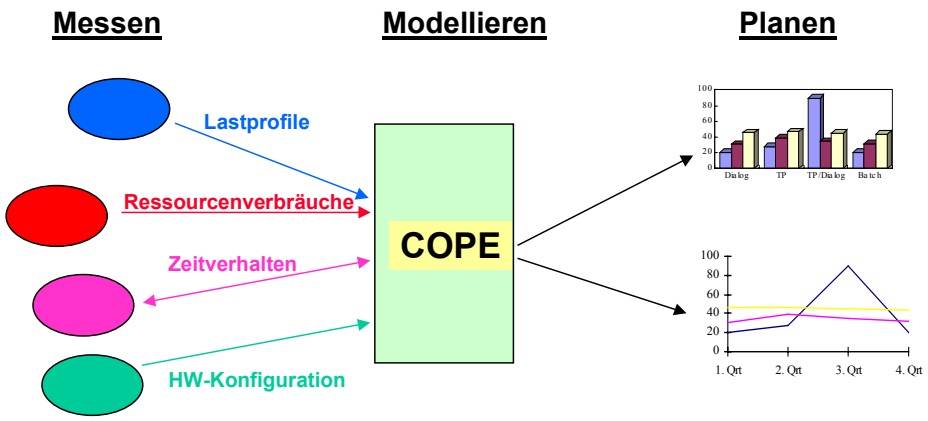


Abbildung 04-12: Ablauf von Messen, Modellieren, Planen

VITO-PERMEX (Universität Essen, Siemens)

VITO ist in der Lage, entsprechend formatierte und konfigurierte Ergebnisse aus PERMEX-Messungen zu importieren und daraus ein Modell des vermessenen Systems zu erstellen. Die Topologie dieses Systems ist in Abbildung 04-13 dargestellt.

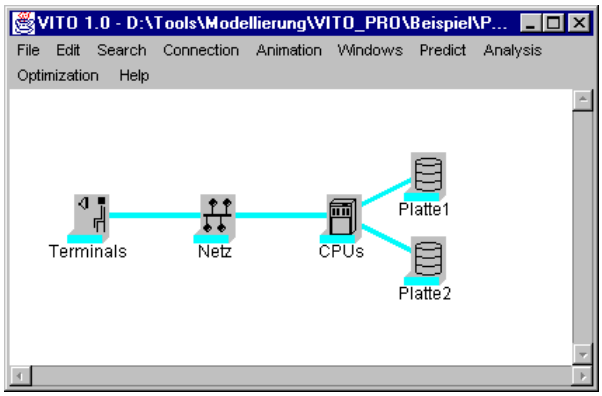


Abbildung 04-13: Aus PERMEX-Daten gewonnenes VITO-Modell

Es werden ein Server mit m Netzinterfaces, mit n CPUs und mit 2*k I/O-Systemen vermessen und entsprechend modelliert. Für einen Import nach VITO stehen mindestens zwei PERMEX-Messungen des gleichen Systems bei unterschiedlichen Lasten zur Verfügung. Auf Grundlage dieser Werte werden die Modellbausteine ausgewählt, und es werden in Abhängigkeit von der Gesamtlast variierende Ressourcenverbräuche einzelner Lasteinheiten identifiziert. Darüber hinaus wird die aktuelle Hauptspeicherauslastung betrachtet und in Prognosemodellen hochgerechnet.

WLPSizer – myAMC (Universität Essen, Siemens)

Der WLPSizer erhält als Eingabe Beschreibungen von R/3-Konfigurationen und Arbeitslasten. Die zur Lastbeschreibung benötigten Informationen liefert das Softwareprodukt myAMC.LNI, welches im Expert Modus Lastbeschreibungen erzeugen kann, aus denen automatisch Workload-Beschreibungen für den WLPSizer gewonnen werden. Die Generierung der Workloads aus den myAMC.LNI-Daten erfolgt mit Hilfe des in den WLPSizer integrierten Programms WLPMaker, das darüber hinaus zur Workload-Analyse genutzt werden kann. Es beinhaltet folgende Features:

- Erstellung von Standard- und Extended-Workloads; Standard-Workloads = Modelleingabe-Daten, bei Extended-Workloads zuzüglich der Kalibrierungsdaten (Antwortzeiten)
- Generierung von Workload-Sequenzen
- Darstellung der Workloads im TCQ-Profil-Format

KAPITEL 05 PRAKTISCHE UMSETZUNG

R. BORDEWISCH, C. FLUES, R. GRABAU, J. HINTELMANN,
K. HIRSCH, F.-J. STEWING

05.01 Erfahrungen bei der KapMan-Prozessetablierung

Die Grundvoraussetzung für das Gelingen eines KapMan-Vorhabens ist die Bestimmung und Schaffung der notwendigen Rahmenbedingungen. Dabei hat sich der KapMan-Einstieg über den Vertrieb als Irrweg erwiesen. Somit ist auch die ursprünglich angestrebte Verankerung von KapMan im Vertriebsprozess nicht sinnvoll. Vielmehr ist der KapMan-Prozess als eigenständiger Serviceprozess im System Life Cycle zu sehen. Der KapMan-Einstieg über die Überwachung eines existierenden IT-Systems, in dem oft bereits Performance-Probleme bestehen, ist in der Praxis am häufigsten realisierbar. Dementsprechend wurde ein erweitertes Vorgehensmodell entwickelt.

Die Erfahrungen und Erkenntnisse, die im Rahmen von MAPKIT bei der Einführung (und Etablierung) des Kapazitätsmanagement für die MAPKIT-Projektanwender LVA Rheinprovinz und RZ Leipzig sowie beim Erprobungsanwender Rechenzentrum der Bundesfinanzverwaltung Frankfurt (RZF) und für ausgewählte R/3-Anwender gewinnen konnten, wurden entsprechend den KapMan-Phasen und KapMan-Methoden ausgewertet. Neben Teilerfolgen bei der LVA Rheinprovinz (Instrumentierung, Vermessung und Prognose-Modellierung einer geschäftskritischen Anwendung, a posteriori Einführung eines Management-Leitstands) und einigen der R/3-Anwendern (Engpassanalyse, Aufzeigen Tuningpotenzial, teilweise auch Prognosemodellierung) ist es beim RZF gelungen, für das Projekt ATLAS ein durchgängiges Kapazitätsmanagement zu etablieren. Der Ablauf der Phasen ist im Detail nachfolgend beschrieben. Eine allgemeine Beschreibung des hier skizzierten Vorgehens in Leitfäden hat sich als nicht praktikabel herausgestellt. Als wichtigster Grund hat sich die sehr große Varianz aller zu beachtenden Einflussgrößen erwiesen. Hierzu zählen u.a. HW/SW-Konfiguration, Spezifika der Last, Performance-Anforderungen.

05.02 Beispiel Projekt ATLAS

Analyse und Tuning: Wie in vielen anderen Projekten erfolgte auch im Projekt ATLAS der KapMan-Einstieg in der Phase Analyse und Tuning. Aufgrund aktueller Performance-Probleme wurden zunächst im Testumfeld und später im Realbetrieb unter Verwendung

der automatisierten Mess- und Auswerteumgebung PERMEX Performance-Messungen durchgeführt mit dem Ziel, Leistungsengpässe zu lokalisieren und Tuning-Möglichkeiten aufzuzeigen. Die Tuning-Maßnahmen betrafen sowohl das Hardware-Upgrading als auch Modifikationen in den Applikationen. Die Auswirkungen dieser Maßnahmen wurden in einigen Massentests überprüft, wo im Testbetrieb 30 bzw. 10 reale User Hochlastsituationen produzierten und die Belastung der Systemkomponenten mittels PERMEX ermittelt und analysiert wurde. Daneben wurden umfangreiche DB-Analysen durchgeführt. Dabei wurden neue Engpässe lokalisiert und Tuning-Empfehlungen erstellt, die in einem Anwendungs-Tuning mündeten.

Überwachung: Aufgrund der Ergebnisse der obigen Performance-Messungen stellte sich die Notwendigkeit zur Überwachung des Systemverhaltens hinsichtlich Performance. Dazu wird PERMEX zur Langzeitüberwachung der Komponenten CPUs, Hauptspeicher, Festplatten und Netzzugang der UNIX-Server-Systeme im Produktivbetrieb eingesetzt. Auf Basis von Wochen- und Monatsstatistiken werden Trendanalysen erstellt, um drohende Engpässe frühzeitig zu erkennen und entsprechende Maßnahmen rechtzeitig einleiten zu können.

Ein anderer Aspekt ist die Einhaltung der garantierten Dienstgüte (Quality of Service) und die Verfügbarkeit, wobei mittels Referenz-PC das Antwortzeitverhalten business-kritischer Transaktionen objektiv ermittelt wird. Dabei wird an einem ausgewählten Client-PC der Dienststelleninfrastruktur ein realer User durch einen automatischen Treiber für die realen Applikationen simuliert. So werden die Antwortzeiten der definierten Transaktionen an der Mensch-Maschine-Schnittstelle angelehnt an DIN 66723 ermittelt und Trendanalysen geliefert. Es werden weiterhin Schwellwerte überwacht und bei Überschreitungen Alerts an den Administrator gemeldet (Kopplung Referenz-PC mit PERMEX).

Planung und Prognose: Im Zuge der ATLAS-Serverkonsolidierung (Übergang von derzeit 25 gleichzeitig aktiven Benutzern auf mehr als 250 an einem Server-System) werden Fragen aufgeworfen, die sowohl funktionaler Art sind als auch Angaben zum Systemverhalten zum Ziel haben:

- Kann ATLAS mehr als 250 Clients handhaben?
- Welche Antwortzeiten sind im WAN- und LAN-Bereich zu erwarten?
- Wie sind die Server-Systeme und das Netzwerk zu dimensionieren?
- Wo sind Engpässe zu erwarten? Wie können diese behoben werden?

Zur Absicherung der Aufnahme eines störungsfreien Produktivbetriebs ist es deshalb erforderlich, einen entsprechenden Last- und Performance-Test durchzuführen. Dabei werden im SBS HSLAN Center in Paderborn 263 Client-PCs und unterschiedliche Server-Systeme

wie DB-Server (Solaris), Kommunikations-, Dienstleistungs- und PDB-Server (Reliant UNIX) mit der realen ATLAS-Anwendung installiert und die LAN-/WAN-Infrastruktur realitätsgetreu nachgebildet. Die Steuerung der Applikationen erfolgt mittels des Treibersystems SQA-Robot.

Es werden unterschiedliche Messwerkzeuge eingesetzt:

- Überwachung der Dienstgüte (Antwortzeiten) am Client mit dem Referenz-PC und SQA-Robot
- Ermittlung der Belastung der Server-Systemkomponenten (CPUs, Hauptspeicher, Disk-I/O-Subsystem, Netzzugang) mit PERMEV (Solaris) bzw. PERMEX (Reliant UNIX)
- Ermittlung der Netzlasten mit verschiedenen Netzwerkanalysatoren

Weiterhin werden die Ansätze zur Test- und Messautomatisierung erprobt.

Ausblick auf Proaktives Kapazitätsmanagement: Realisierung: die Kopplung von PERMEV (bzw. PERMEX) mit dem Referenz-PC bietet die Vorteile einer durchgängigen Überwachung der beteiligten Software-Komponenten:

- Referenz-PC und ADAX (Alert-Dämon auf UNIX) liefern regelmäßige Alive-Messages an die Überwachungseinheit UWE
- Alerting von Statusmeldungen (Resultat der durchgeführten Transaktion) erfolgen abhängig vom jeweiligen Ergebnis in unregelmäßigen Abständen
- Der Online-Auswerter überwacht die aktuellen Serverressourcen (Kurzfassung) und sendet ebenfalls an UWE

Je nach Anzahl bereits erfolgter SLA-Verletzungen wird ein gestuftes Alarmsystem aktiviert. Durch die Referenz-PC/PERMEX-Kopplung wird ein Vergleich der Trendanalysen von Server-System und Client ermöglicht. Weiterhin werden bei Überschreitungen der Schwellwerte gezielt detaillierte Engpass- und Ursachenanalysen mittels PERMEX durchgeführt.

Um frühzeitig Maßnahmen einleiten zu können, bevor sich ein Anwender beschwert, wird das proaktive Kapazitätsmanagement auf Basis der Kopplung Referenz-PC mit den PERMEX-Langzeitmessungen durchgeführt mit dem Ziel: Von der reinen Problemerkennung hin zur Problemerkennung und Problemvermeidung. Unbeschadet von aktuellen Auswertungen besteht somit die Möglichkeit, mit kurzer Reaktionszeit die Ursachenanalyse zu starten. „Resource on Demand“ wird zunächst als Definition und Umsetzung von organisa-

torischen Maßnahmen verstanden. Deren Steuerung ist wie die Integration in ein Framework Bestandteil der Erweiterungen dieses Ansatzes.

05.03 KapMan-Training und Weiterqualifikation

Es werden für unterschiedliche Adressatenkreise Workshops angeboten, die sich einerseits an Fachleuten orientieren und auf fachspezifische Themen näher eingehen, wie z.B. "Performance-Betrachtungen in geswitchten IT-Systemen" mit dem Fokus auf Analysen und Bewertungen des Verhaltens des Gesamtsystems aus Endbenutzersicht, und die andererseits dazu dienen, Managern und anderen Interessierten einen KapMan-Überblick zu verschaffen.

Oft können Anfragen mit frei verfügbaren Projektergebnissen beantwortet werden. Dadurch entstehen Kooperationen im Bereich der Forschung und Entwicklung, aber auch mit Anwendern und potentiellen Kunden.

05.04 Vorbereitung zum Erstgespräch bei Anwendern

Die folgende Struktur liefert einen Orientierungsrahmen zum Erstgespräch und soll verdeutlichen:

@ durchzuführende Arbeitsschritte bei/mit Kunden

@ in den verschiedenen Phasen einzuholende Informationen

Zu den folgenden Punkten 1 bis 4 gibt es im ächsten Abschnitt eine detailliertere Fragensammlung. Diese ist zur Vorbereitung von Kundenworkshops den Teilnehmern auszuhändigen.

1. Ermittlung Kundenprofil

@ Allgemeine Informationen über IT-Betrieb

@ Branchenzugehörigkeit

@ Bedeutung der IT für das Unternehmen/Betreiber/Kunden

@ Aktuelle Probleme

@ Randbedingungen

2. Bestandsaufnahme Ist-Situation

@ Welche wesentlichen Geschäftsprozesse gibt es heute?

@ Wie lassen sich diese Geschäftsprozesse auf IT-Prozesse (Verfahren) abbilden?

@ Welche Ressourcen HW- und SW-seitig beanspruchen die IT-Prozesse mit welcher Intensität? *Stichworte:* Server, Clients, Netzkomponenten, Applikationen, Datenbanken, Mengengerüste

@ Welche Anforderungen werden an IT-Prozesse gestellt bzw. lassen sich daraus für den IT-Betrieb ableiten? *Stichworte:* Servicegüte, Verfügbarkeit, Ausfallsicherheit, Administration und Überwachung, Automatisierung. *Hilfsmittel:* Übersichtspläne HW- und SW-Konfiguration, Zuordnung HW/SW zu Verfahren

3. Sollkonzept - Festlegung des erwarteten oder geplanten Zustandes

@ Welcher Personenkreis ist im Planungs- und Realisierungsprozess eingebunden? Ansprechpartner bei:

- IT-Leitung
- Systemseite (Applikation, Datenbank, Netz)
- Fachbereiche (Erfordernisse aus Sicht des Tagesgeschäftes)

@ Wie lassen sich die vorgesehenen Maßnahmen in einem Stufenplan zuordnen:

- kurzfristige Ziele 1 - 3 Monate
- mittelfristige Ziele 3 Monate - 1 Jahr
- langfristige Ziele 1 - 3 Jahre

@ Was sind die möglichen Auswirkungen der geplanten Änderungen auf die Organisation und somit auf Struktur/Ablauf der Geschäftsprozesse und der involvierten IT-Prozesse?

- Neue Verfahren
- Ausweitung oder Abschaffung bestehender Verfahren
- Umsetzung oder Neukonzeption existierender Verfahren
- Plattformwechsel

Die Bestandsaufnahme zum Sollkonzept läut im Wesentlichen analog zu Punkt 2.

4. Bewertung von Lösungsvarianten

@ Gegenüberstellung von Konfigurationsalternativen

@ Diskussion von Vor- und Nachteilen unter verschiedensten Aspekten
  - Charakteristika von HW- und SW-Architektur
  - Performance
  - Verfügbarkeit
  - Kosten
@ Administration (Netz, Server, Clients)

05.05 Detaillierte Fragensammlung

1. Ermittlung Kundenprofil

@ Zu welcher Branche gehört der IT-Betreiber bzw. gehören die Kunden des IT-Betreibers?

@ Was sind die globalen geschäftlichen Ziele?
  - Kostenverringerung
  - Produktivitätsverbesserung
  - Kundenzufriedenheit
  - Mitarbeiterzufriedenheit
  - Informationsverteilung
  - Außendarstellung

@ Gibt es allgemeine Vorgaben?
  - Infrastruktur
  - Etat
  - ...

@ Wer berät den IT-Betreiber?

2. Bestandsaufnahme Ist-Situation

@ Welche wesentlichen Geschäftsprozesse (GP) gibt es heute?
  - Auskunft

- Einspruch, Beschwerde
- Rechnungsstellung
- ...

@ Wie lassen sich diese GPs auf IT-Prozesse abbilden?

- Transaktionsnamen
- Programm- oder Jobnamen (Tasks oder Prozesse)
- Ereignisse für Beginn und Ende eines GP oder eines Teilschritts eines GP

@ Welche IT-Services nehmen die IT-Prozesse (Transaktionen) in Anspruch?

- Applikation
- Datenbank
- Transport
- Verwaltungsdienste (Authorisierung, Lock-Manager,Inventarisierung)
- systemtechnische Basisdienste (Meßwerterfassung, Statistiken)

@ Welche IT-Komponenten (CPU, Platten, Netz) nehmen die IT-Services in Anspruch?

- Wie hoch ist der Ressorcenverbrauch an den wesentlichen IT-Komponenten?
- Wie können die Werte erhoben werden?
- Wie lassen sich die Verbrauchswerte den verursachenden Geschäftsprozessen zuordnen?

@ Wie hoch sind die operationalen Kosten?

- HOST-CPU, IO
- Netz, Server
- Database-Handler

@ Mit welcher Häufigkeit treten diese Geschäftsprozesse auf?

- Absolut
- Relativ zueinander

- In Bezug auf zugehörige IT-Prozesse oder vorgeschaltete Instanzen (Anzahl User, Anzahl Kunden, Konten, ...)
- Welche periodischen Abhängigkeiten lassen sich erkennen bei unterschiedlichen Bezugszeiträumen, wie Tag, Woche, Monat, Jahr?
- Welche IT-Prozesse sind zeitkritisch, welche können in Schwachlastphasen verlegt werden?

@ Welche Servicegüteforderungen werden gestellt?

- Vorgangszeit
- Antwortzeit (Stabilität, absolute Antwortzeit, relativ zur Komplexität)
- Durchsatz
- Verfügbarkeit
- Vorgangskosten
- Qualität der angebotenen Dienste

@ Wie kann Servicegüte gemessen werden?

- an der Komponente
- für den IT-Prozess
- für den Geschäftsprozess

@ Wie ist der IT-Betrieb organisiert?

- Bediente und unbediente Betriebszeiten
- Verfügbarkeitsanforderungen, Backup-Konzepte
- Administration
- Automatisierung

@ Gibt es Übersichten über vorhandenes Inventar?

- Server-Konfiguration
- Netzübersicht
- SW-Lizenzen und -Stände

3. Sollkonzept - Festlegung des erwarteten oder geplanten Zustandes

@ Welcher Personenkreis ist im Planungs- und Realisierungsprozess eingebunden? Welches sind die Ansprechpartner bei:

- IT-Leitung
- Systemseite (Applikation, Datenbank, Netz)
- Fachbereiche (Erfordernisse aus Sicht des Tagesgeschäftes)

@ Wie lassen sich die vorgesehenen Maßnahmen in einem Stufenplan zuordnen:

- kurzfristige Ziele 1 - 3 Monate
- mittelfristige Ziele 3 Monate - 1 Jahr
- langfristige Ziele 1 - 3 Jahre

@ Was sind die möglichen Auswirkungen der geplanten Änderungen auf die Organisation und somit auf Struktur/Ablauf der Geschäftsprozesse und der involvierten IT-Prozesse?

- Neue Verfahren
- Ausweitung oder Abschaffung bestehender Verfahren
- Umsetzung oder Neukonzeption existierender Verfahren
- Plattformwechsel

Im Wesentlichen ist bei der Ermittlung des Sollkonzepts ebenfalls sinngemäß der Fragenkatalog aus Punkt 2 anzuwenden!

4. Bewertung von Lösungsalternativen

@ Gegenüberstellung und ggf. Modellierung von Konfigurationsalternativen:

- Verteiltes System
- Zentrales System

@ Diskussion von Vor- und Nachteilen unter den Aspekten:

- Verfügbarkeit, Ausfallsicherheit, Notbetrieb, Backup
- Konnektivität, Schnittstellen

- Wartbarkeit
- Performance (Antwortzeiten: Host, Netz)
- Betriebskosten (Netzkosten, Personal)
- Anschaffungskosten
- Einsatz wesentlicher HW- und SW-Komponenten
- Mono- oder Multiprozessor, SMP, MPP
- Cachingstrategien
- DB-System (Mono- oder Multitask, Multithread, ...)
- TA-Monitor
- Vernetzungsbausteine (HW, SW)

@ Besonderheiten von HW und SW (Applikationen):

- Funktionalität
- Verbrauchsverhalten
- Verträglichkeit bzw. Kompatibilität verschiedener Komponenten
- Restriktionen/Obergrenzen (z.B. Anzahl möglicher Verbindungen, Partner, Tasks, ...)

@ Administration (Netz, Server, Clients):

- Datensicherung
- SW-Verteilung
- Automatisierung

KAPITEL 06 HERAUSFORDERUNGEN UND CHANCEN

KATHRIN SYDOW

06.01 Umgang mit komplexen Situationen im IT-Bereich

Zur Einstimmung in das Thema werden Exzerpte aus dem Buch von Dietrich Dörner ‚Die Logik des Misslingens' (Rowohlt Verlag, Reinbeck, 1992) vorangestellt.

(a) Komplexität

... Die Existenz von vielen, voneinander abhängigen Merkmalen in einem Ausschnitt der Realität wollen wir als "Komplexität" bezeichnen. Die Komplexität eines Realitätsausschnittes ist also umso höher, je mehr Merkmale vorhanden sind und je mehr diese voneinander abhängig sind (entscheidend ist auch die Qualität/Art der Beziehungen) ....

... Eine hohe Komplexität stellt hohe Anforderungen an die Fähigkeit eines Akteurs, Informationen zu sammeln, zu integrieren und Handlungen zu planen. Nicht die Existenz vieler Merkmale allein macht die Komplexität aus. ... Erst die Vernetztheit, also die zwischen den Variablen des Systems existierenden Verknüpfungen, macht die gleichzeitige Betrachtung sehr vieler Merkmale notwendig und bringt es mit sich, dass man in solchen Realitätsausschnitten fast nie nur eine Sache machen kann. ... Ein Eingriff, der einen Teil eines Systems betrifft oder betreffen soll (!), wirkt immer auch auf viele andere Teile des Systems ...

(b) Dynamik

... Realitätsausschnitte sind nicht passiv, sondern – in gewissem Maße – aktiv. Dies erzeugt zum Beispiel Zeitdruck: man kann nicht "ewig" warten, bis man sich schließlich zu einem Eingriff entschließt ... Die Eigendynamik von Systemen macht weiterhin die Erfassung ihrer Entwicklungstendenzen bedeutsam. Bei einem dynamischen Gebilde darf man sich nicht damit zufriedengeben zu erfassen, was der Fall ist. Die Analyse der Gegebenheiten reicht keineswegs aus. Man muss zusätzlich versuchen herauszubekommen, wo das Ganze hinwill ...

(c) Intransparenz

... Ein weiteres Merkmal ... ist die Intransparenz von Situationen. ... Es ist nicht alles sichtbar, was man eigentlich sehen will. ... Zusammengefasst: Viele Merkmale der Situation

sind demjenigen, der zu planen hat, der Entscheidungen zu treffen hat, gar nicht oder nicht unmittelbar zugänglich ...

(d) Unkenntnis und falsche Hypothesen

... das Realitätsmodell eines Akteurs ... (also seine implizite, persönliche Sichtweise der Situation) ... kann ... richtig oder falsch sein, vollständig oder unvollständig sein. Gewöhnlich dürfte es sowohl unvollständig als auch falsch sein, und man tut gut daran, sich auf diese Möglichkeit einzustellen ... Menschen streben nach Sicherheit – das ist eine der (Halb-)Wahrheiten der Psychologie ... und dieses Streben hindert sie, die Möglichkeit der Falschheit von Annahmen oder die Möglichkeit der Unvollständigkeit angemessen in Rechnung zu stellen ...

(e) Stationen des Planen und Handelns

Eine Abbildung in dem Buch von Dietrich Dörner (Abb. 17) zeigt den logischen Ablauf mit Rücksprüngen. Er schreibt dazu.

... So ungefähr könnte man sich die Stationen der Organisation komplexen Handelns vorstellen. Natürlich ist der Weg, der in der Abbildung schematisch dargestellt ist, gewöhnlich nicht ein einfaches Fortschreiten von Station zu Station. ... wie in der Abbildung dargestellt, gibt es von jeder Station zu jeder Station des Handlungsweges Rücksprünge. So kann die tatsächliche Planung eines komplizierten Maßnahmenpakets aus einem vielfältigen Hin- und Herspringen zwischen diesen verschiedenen Stationen bestehen ... Die Stationen in der Abbildung enthalten die Probleme, die gelöst werden müssen ...

(f) Überleitung zum Kapazitätsmanagement von IT-Systemen

Trotz der zunehmenden Abhängigkeit des Unternehmenserfolgs von der Leistungsfähigkeit und der Verlässlichkeit immer vielschichtiger werdender IT-Strukturen ist eine systematische Betrachtung des Gesamtsystemverhaltens in vielen Unternehmen völlig unzureichend etabliert.

Der Umgang mit komplexen Situationen im IT-Bereich wird jedoch täglich unausweichlicher.

Es kommt also im ersten Schritt darauf an, das Unbehagen angesichts steigender Komplexität zu überwinden und sich der Situation zu stellen. Als nächstes müssen methodische Ansätze erkannt und angewendet werden, die die Komplexität begreifbarer und relevante Steuer- und Einflussgrößen deutlicher oder überhaupt erst sichtbar machen. Im Bereich Leistungsverhalten und Kapazitätsbedarf gibt es hier den methodischen Ansatz des Kapazitätsmanagements, der im Folgenden erläutert wird.

06.02 Was ist Kapazitätsmanagement?

Zuerst sei erwähnt, dass Kapazitätsmanagement in den hier beschriebenen Zusammenhängen branchenneutral und plattformübergreifend betrachtet wird. Es geht also nicht um einzelne Tools oder Verfahren, sondern um ein Prinzip, das mittels eines Gesamtkonzepts umgesetzt werden muss.

Grundsätzlich handelt es sich bei Kapazitätsmanagement um Themen, die gemeinhin höchst ungenau mit dem Begriff Performance umschrieben werden. Jeder versteht in dem Bereich, in dem er sich befindet, nämlich etwas anderes darunter.

Das Spektrum reicht von Anzahl Paketen und Kollisionen auf Netzstrecken über Ressourcenverbräuche auf Serverebene bis zur Gesamtsystemleistung, die sich z.B. in Reaktionszeiten, Laufzeiten und letztendlich zu erfüllenden Geschäftszielen manifestiert.

Hinzu kommen - an dieser Stelle selten betrachtete - Gesichtspunkte, nämlich allgemeine Rahmenbedingungen und Einflüsse organisatorischer, vertrieblicher, ja sogar politischer Art, Investitions- und Planungszyklen, die Argumentation von Investitionen aufgrund der Leistungsanforderungen und die Wechselwirkungen zwischen Geschäfts- und IT-Prozessen.

Je komplexer, vielfältiger und ‚mission critical' die Anforderungen aus Geschäftsprozessen sind, desto sorgsamer muss die sinnvolle Umsetzung in leistungsfähige IT-Prozesse erfolgen, muss die fortlaufende Betrachtung gewährleistet sein.

Je heterogener und verteilter eine IT-Landschaft ist, desto höher ist die Zahl beeinflussender Faktoren auf die Leistungsfähigkeit der IT-Prozesse. Hinzu kommt, dass es sich nicht um statische, sondern sich permanent ändernde, dynamische, ineinander greifende Situationen handelt, die es zu begreifen und zu beschreiben gilt.

Das Prinzip Kapazitätsmanagement ist also auf allen Ebenen, vom Geschäftsprozess bis zur installierten Hardwarekomponente und somit für alle betroffenen Personen relevant; und dies nicht zeitlich begrenzt, sondern fortlaufend.

Im laufenden Betrieb ist reibungsloser, performanter Ablauf zu garantieren, auftretende Leistungsprobleme müssen schnellst- und bestmöglich gelöst werden, Eingangsdaten für anfallende Ressourcenplanungen, Leistungsanforderungen und Entscheidungen sind zu finden.

06.03 Bridging the Gap – mehr als Worte

Ist es überhaupt sinnvoll, einen Weg zu beschreiten, der mich vom Geschäftsprozess bis in die Tiefen der Bits und Bytes und wieder zurück führt?

Nach unseren bisherigen Erfahrungen ist es sogar erforderlich, wenn man jedes relevante Ereignis im Kontext der Umgebung betrachten muss, um Ursache und Wirkung zu ergründen.

Althergebrachtes, klassisches Kapazitätsmanagement und das Förderprojekt MAPKIT beschreiben den Weg auf interessante Weise. Während klassisches Kapazitätsmanagement meist mit der Erfassung von Messdaten auf Hardware- oder Softwareebene beginnt und erst der Prognostiker / Modellierer nach Kenndaten wie Anwenderverhalten und Planungsdaten fragt, startet MAPKIT in der ersten Stufe mit Workshops aller Beteiligten, um Rahmenbedingungen, Kernprozesse, Planungsstrategien zu erforschen.

Erst in zweiter Stufe erfolgt die Analyse der IST-Situation nach festgelegten Kriterien.

Auf Basis aller Erkenntnisse werden Aktionspläne aufgestellt, die ein bestimmtes Zeitraster abdecken und kurz-, mittel- und langfristige Aspekte berücksichtigen. Erst innerhalb der Aktionspläne treten Aktivitäten auf technischer Ebene auf, deren Ergebnisse wiederum in Stufe 1 und 2 einfließen. Der hier entwickelte methodische Ansatz ermöglicht eine Standortbestimmung, die jedes weitere Vorgehen erleichtert oder überhaupt erst möglich macht.

Hier wird auch die Position von Kapazitätsmanagement selbst klar. Aufgabe von Kapazitätsmanagement ist es nicht, Geschäftsprozesse zu gestalten, sondern sie zu verstehen sowie Auswirkungen des IT-Umfelds rechtzeitig zu erkennen und die Erkenntnisse zu nutzen.

Die Funktionalität und das Zusammenspiel von Soft- und Hardwarekomponenten wird von Kapazitätsmanagement bezüglich Leistungsverhalten / Servicegüte / Ressourcenbedarf etc. kritisch betrachtet und mit Empfehlungen kommentiert, jedoch nicht entworfen, entwickelt oder gewartet.

So stellt sich Kapazitätsmanagement sowohl als Informationsgeber als auch als –nutzer dar.

06.04 Vom Prinzip zum Handeln

Ist es möglich, alle oder wenigstens die wichtigen Facetten des Kapazitätsmanagements in Vorgehensweisen abzubilden? Wie unsere Erfahrung aus zahlreichen Aktivitäten und Projekten gezeigt hat, ist dies grundsätzlich bereits heute machbar.

Wesentliche Punkte sind dabei:

- Absprache mit involvierten Bereichen
- Beratung der IT-Bereiche auf Basis der Geschäftsprozesse
- Erfassen der Ist-Situation
- Erfassen relevanter strategischer Überlegungen im IT-Bereich
- Entwickeln des erforderlichen Sollkonzepts
- Einschätzung und Koordination im Projekt: Know-how, Methodik, Tools, Zeitpunkt
- Umsetzung erforderlicher Kapazitäsmanagement-Aktionen
- Präsentation, Argumentation
- Vertragsgestaltungen hinsichtlich Leistungsanforderungen (SLAs)

Selbstverständlich muss für jeden IT-Bereich und jedes Projekt die geeignete Vorgehensweise zusammengestellt werden, die auch den jeweiligen Erfordernissen entspricht.

Außerdem muss Wert darauf gelegt werden, alle Beteiligten entsprechend einzubinden und zu informieren. Dies bedeutet zum Beispiel, mit dem jeweiligen IT-Anwender sehr frühzeitig zu erarbeiten, wie entscheidend das Verhalten seiner End User auf Systemverhalten und Ressourcenbedarf sein kann.

Das kann auch bedeuten, in Fällen von Ausschreibungen / Anschaffungen mit dem zuständigen IT-Anbieter und dem IT-Anwender zu überlegen, ob Leistungsmerkmale in der geforderten Form realistisch und überprüfbar sind, ob es brauchbare Referenzen gibt, ob der IT-Anwender alle erforderlichen Eingangsdaten liefern kann und ob gegebenenfalls Stufenkonzepte sinnvoll sind.

06.05 Kapazitätsmanagement als Retter in der Not?

Wie die Praxis zeigt, wird Kapazitätsmanagement nach wie vor oft erst eingesetzt, wenn massive Leistungsprobleme auftauchen und das Behandeln von Symptomen keine Besserung mehr bringt. Meist fehlt es dann am erforderlichen Budget, die Zeit ist knapp, kostspielige Nachrüstungen und Imageverlust drohen, der negative Einfluss auf die Geschäftsprozesse wird überdeutlich.

Die Beteiligten im IT-Bereich, an der Ausschreibung oder dem Projekt sind bereits seit Wochen, Monaten oder vielleicht sogar seit Jahren involviert.

Ein erst im Problemfall aktivierter Kapazitätsmanagement-Experte soll nun aber in kürzester Zeit die Materie erforschen und verstehen, Zusammenhänge sehen, die richtigen Aktionen starten und natürlich die Probleme so schnell und so billig wie möglich lösen.

Lösung kann hier eigentlich nur sein, dass das Thema Kapazitätsmanagement und der Umgang damit zum Basiswissen jedes Vertriebs, jedes Projektmanagements gehört sowie zum täglichen Umgang mit IT-Themen, dass Leistungsverhalten und Kapazitätsbedarf in der Prioritätenliste des Tagesgeschäfts, des Projektmanagements, des Vertriebs nicht erst hinter Problembehebung, Projektübergabe oder Angebotsabschluss steht, dass frühzeitig entsprechende Zeiträume und Budgets eingeplant und erforderliche Spezialisten involviert werden.

06.06 Die täglichen Hürden – zwischen Begeisterung und Ablehnung

Die Reaktionen und Kommentare zu Kapazitätsmanagement sind sehr unterschiedlich. Die Verlautbarungen reichen von ‚endlich kümmert sich jemand darum' bis ‚das braucht doch kein Mensch'.

Tatsache ist, wer aus akuter Problematik heraus Kapazitätsmanagement einsetzte, konnte seine Position in der Regel verbessern, auf stabilerer und sachlicherer Argumentationsbasis weiterarbeiten, eine transparentere Situation erzeugen. Dies trifft sowohl auf die IT-Anwender als auch Projektmanager und Vertriebsmitarbeiter zu. Auch der entscheidende Einfluss bei Vertragsformulierungen sollte für beide Seiten nicht unterschätzt werden.

Tatsache ist jedoch, dass nur wenige bereit sind, von Anfang an das Thema Kapazitätsmanagement in ihren Planungen und Budgets zu berücksichtigen. Die Gründe mögen vielfältig sein, die Wirkung ist jedoch fatal.

Gerade spezielles Know-how, wie es für Kapazitätsmanagement erforderlich ist, verkümmert zwangsläufig, wenn keine regelmäßige Inanspruchnahme und Ausbildung stattfindet.

Nahezu jedes Thema muss heutzutage über Business- oder Kostenpläne und deren Einhaltung gerechtfertigt werden. Es ist dadurch sowohl beim IT-Anbieter als auch beim IT-Anwender nahezu unmöglich geworden, erforderliche Investitionen zu argumentieren, die sich auf branchen- und abteilungsübergreifende Belange beziehen sowie keinen schnellen Profit versprechen, sondern für prophylaktisches Verhalten zur Aufrechterhaltung kritischer Geschäftsprozesse erforderlich sind.

So wird es in Zukunft – wenn sich die Situation nicht ändert - immer häufiger vorkommen, dass in extrem schwierigen Situationen, die den erfahrenen Kapazitätsmanagement-Experten erfordern, dieser nicht mehr zur Verfügung steht.

Experten, die den Anforderungen des Kapazitätsmanagements entsprechen, also zu systemischen und vernetzten Betrachtungen in der Lage sind, sind lernfähig, flexibel und belastbar und werden deshalb andere Perspektiven wahrnehmen.

Dies ist gleichermaßen verhängnisvoll für die sichere Abwicklung kritischer Geschäftsprozesse, für nicht professionell genug betreute Aufträge und Projekte und letztendlich auch für das Know-how-Profil eines Unternehmens.

06.07 Keine leichte Aufgabe

Wer also Kapazitätsmanagement ernsthaft betreiben will, muss sich mit ausreichend Beharrlichkeit und Zähigkeit wappnen.

Viel Basisarbeit ist zu leisten; es gilt nicht nur, Mitarbeiter und Kollegen zu überzeugen und ihr Vertrauen und Interesse zu gewinnen, sondern über das heute verfügbare Know-how hinaus weitere Fachleute mit entsprechendem Potenzial zu finden und diese zu schulen, zu motivieren und in Projekten gezielt einzusetzen.

Da speziell die für komplexere IT-Landschaften und Aufgabenstellungen erforderlichen Fachleute in der Regel in einem Unternehmen in den unterschiedlichsten Bereichen, in anderen Unternehmen (z.B. bei den IT-Anbietern) sowie Universitäten zu finden sind, ist der Aufbau eines Netzwerks erforderlich.

Vorhandenes und entstehendes Know-how muss sinnvoll verdichtet werden, gemeinsame Projekte und Projektteams müssen gegebenenfalls betreut und gesteuert werden.

Da es sich um Know-how handelt, das branchen- und plattformübergreifend, in Hardware- ebenso wie in Lösungsbereichen, in der Anwendungsentwicklung wie in kleinen, mittleren und Großprojekten eingesetzt werden muss, erscheint eine Kooperation und gegenseitige Beteiligung und Förderung gerechtfertigt und sinnvoll.

06.08 Virtuelles Team – nur ein frommer Wunsch?

Es ist sicherlich ein hoher Anspruch, ein Virtuelles Team über lose Zusammentreffen fachlicher Art hinaus in ein funktionierendes und sozusagen geschäftstüchtiges Gebilde zu verwandeln.

Die Fähigkeit, über die absolute Abgrenzung und Konkurrenz hinaus mit anderen Bereichen und Unternehmen in sinnvolle Kooperation zu treten, ist aber oft erforderlich, um überhaupt akzeptables und profitables Geschäft betreiben zu können.

Gerade im Bereich des Kapazitätsmanagements ist Know-how in beliebiger Ausprägung über das Gesamtunternehmen und darüber hinaus verstreut.

Um den Erfordernissen eines profitablen Geschäfts zu genügen, sollte aber in jeder Situation, in der Kapazitätsmanagement erforderlich ist, das entsprechende Know-how bekannt sein und ebenso sinnvoll wie schnell gekoppelt werden können.

Unterschiedliches Verständnis desselben Themas, also unterschiedliche Vorgehensweisen und damit uneinheitliche Ergebnisse erzeugen ungeheure Reibungsverluste, ebenso Konkurrenzdenken, unterschiedliche Modalitäten, Kosten- und Kompetenzgerangel.

Entscheidend für die Bildung eines funktionsfähigen virtuellen Teams ist also der zusammenführende Effekt einer neutralen, gemeinsamen Basis, die die unterschiedlichen Interessen zusammenführt und harmonisiert.

06.09 Kapazitätsmanagement und das Förderprojekt MAPKIT – esoterische Spielerei oder notwendige Innovation?

Die Antwort auf diese Frage ist verhältnismäßig simpel.

In abgegrenzten Bereichen und bekanntem Terrain können mit heutigen Mitteln des Kapazitätsmanagements geschäftskritische Leistungseinbrüche und die damit verbundenen Folgen vermieden werden. Es kann seriöse Konzeptsicherung betrieben und eine einigermaßen sinnvolle Vereinbarung von Leistungsmerkmalen getroffen werden.

Ohne ein Förderprojekt wie MAPKIT, das sich gezielt mit der Bewältigung komplexer Problemstellungen im Bereich Leistung und Servicegüte befasst (unter Einbeziehung neuer Technologien und Anforderungen), ist dies in absehbarer Zeit nicht mehr möglich, weil hierfür das durchgängige Konzept noch entwickelt werden muss.

Unter diesem Aspekt wird weder das Projekt MAPKIT noch die professionelle Umsetzung von Kapazitätsmanagement je so viel kosten wie der Einbruch bei Glaubwürdigkeit, Professionalität und Seriosität, ganz zu schweigen von geschäftsschädigenden Leistungseinbrüchen, die wegen mangelnder systematischer Betrachtung zwangsläufig anfallen.

Teil II: Messung und Monitoring

Radioteleskop Raisting

Inhalt von Teil II

KAPITEL 01 DAS MESS- UND AUSWERTESYSTEM PERMEV 4

01.01 EINFÜHRUNG 4
01.02 DAS MESS- UND AUSWERTESYSTEM 5

KAPITEL 02 PROAKTIVES KAPAZITÄTSMANAGEMENT 10

02.01 EINFÜHRUNG UND MOTIVATION 10
02.02 IM BRENNPUNKT: GESAMTSYSTEMVERHALTEN AUS ENDBENUTZERSICHT 10
02.03 AUTOMATISIERTE MESSUNGEN MITTELS PERMEV 12
02.04 PROAKTIVES KAPAZITÄTSMANAGEMENT 13
02.05 ONLINE-AUSWERTUNG DER SERVERRESSOURCEN 15
02.06 ANWENDUNGSGEBIETE 15

KAPITEL 03 APPLICATION EXPERT 17

03.01 ÜBERSICHT 17
03.02 INSTRUMENTIERUNG UND ANALYSEMETHODEN 18
03.03 BEISPIEL: MEHRSTUFIGE CLIENT/SERVER-KOMMUNIKATION 18
03.04 ABGRENZUNG ZU ANDEREN MESSVERFAHREN 19

KAPITEL 04 SERVICE LEVEL MANAGEMENT 21

04.01 EINLEITUNG 21
04.02 FAZIT 21
04.03 AMDAHL ENVIEW 22
04.04 BMC SITEANGEL 23
04.05 CANDLE EBA*SERVICEMONITOR 25
04.06 HP OPENVIEW VANTAGE POINTS INTERNET SERVICES 26
04.07 KEYNOTE PERSPECTIVE 28
04.08 MERCURY INTERACTIVE TOPAZ 28
04.09 SERVICE METRICS WEBPOINT 29
04.10 TIVOLI WEB SERVICE MANAGER UND WEB SERVICE ANALYSER 30
04.11 ABKÜRZUNGEN 31
04.12 LINKS 32

KAPITEL 05 UNIVERSELLER LASTGENERATOR *SYLAGEN* 33

05.01 EINFÜHRUNG 33
05.02 LÖSUNGSKONZEPT *SYLAGEN* 33
05.03 SYLAGEN FRAMEWORK 34
05.04 SYLAGEN LASTADAPTOR 35

KAPITEL 06 ANFORDERUNGEN AN MESSWERKZEUGE.....37

06.01 EINFÜHRUNG.....37
06.02 ALLGEMEINE ANFORDERUNGEN.....37
06.03 WEITERE ASPEKTE.....39
06.04 SPEZIELLE ANFORDERUNGEN HINSICHTLICH ÜBERWACHUNG.....40
06.05 SPEZIELLE ANFORDERUNGEN HINSICHTLICH ANALYSE UND TUNING.....41
06.06 SPEZIELLE ANFORDERUNGEN HINSICHTLICH PROGNOSE.....42

KAPITEL 01 DAS MESS- UND AUSWERTESYSTEM PERMEV

BÄRBEL SCHWÄRMER

01.01 Einführung

UNIX-basierte Server-Betriebssysteme bieten von Haus aus eine Reihe von Monitoring-Werkzeugen, die es erlauben, dem Server-Verhalten auf die Spur zu kommen. Der bekannteste Vertreter ist der sar (System Activity Reporter), der eine breite und vielfältige Performance-Analyse der wichtigsten Systemkomponenten (CPU, RAM, Festplatten) und Systemfunktionen (z.B. Systemaufrufe, Paging, Swapping) ermöglicht. Das Kommando mpstat dient zusätzlich zur Aufnahme von Multi-Prozessor-Statistiken. Weiterhin wird sehr häufig ps (process state) und acct (accounting) zur Ermittlung Prozess-spezifischer Daten eingesetzt. Anhand der Kommandos ipcs und df können Daten über die Ressourcen der Interprozesskommunikation (z.B. Semaphore) und die Belegung der Filesysteme ermittelt werden. Das Kommando netstat protokolliert die erhaltenen und gesendeten Netz-Pakete und unterstützt die Performance-Analyse des Netzwerks. Darüber hinaus ermöglicht bei einigen UNIX-Betriebssystemen (z.B. LINUX) das virtuelle /proc-Filesystem den Zugriff auf eine Menge von Systemdaten einschließlich prozessspezifischer Daten. Der Nachteil all dieser Monitoring-Werkzeuge besteht darin, dass sie nur unabhängig voneinander zum Zuge kommen können.

Wünschenswert ist eine automatisierte Mess- und Auswertungsumgebung, welche die Handhabung dieser Werkzeuge vereinfacht, die Aufrufe standardisiert und synchronisiert, die erhobenen Performance-Rohdaten über alle Erfassungsquellen korreliert und auswertet sowie insgesamt den Aufwand für Messung und Auswertung in Grenzen hält. Für den Einsatz auf unterschiedlichen UNIX-Server-Plattformen hat Siemens Business Services (SBS) das System PERMEV entwickelt. PERMEV (PERformance Monitoring and Evaluation EnVironment) ist eine Mess- und Auswerteumgebung für Systeme mit UNIX-basiertem Betriebssystem, wie Reliant UNIX, SOLARIS und LINUX.

PERMEV besteht aus den Komponenten

- Messsystem ‚monitor'
- Auswertesystem ‚evaluation'
- Präsentationssystem ‚statistics'

Das Messsystem ‚monitor' sammelt Performance-Daten auf dem zu vermessenden System. Es handelt sich dabei um Daten über die Belegung der Systemressourcen wie CPU, Hauptspeicher, Festplatten und Netzzugang. Diese Daten werden innerhalb der Auswerteumgebung durch die Komponente ‚evaluation' verdichtet und tabellarisch zusammengefasst, wobei das Auswertesystem hinsichtlich der auszuwertenden Performance-Daten variabel konfigurierbar ist. Die durch das Auswertesystem erzeugten Ergebnisdaten werden nachfolgend durch die Komponente ‚statistics' statistisch weiter ausgewertet und aufbereitet. Hierzu gehört die Ermittlung von Mittel- und Maximalwerten, die Berechnung von Korrelationskoeffizienten sowie die grafische Präsentation.

Im Folgenden werden das Messsystem ‚monitor', das Auswertesystem ‚evaluation' und das Präsentationssystem ‚statistics' beschrieben.

01.02 Das Mess- und Auswertesystem

(a) Übersicht Gesamtsystem PERMEV

Die nachfolgende Abbildung 01-1 gibt einen groben Überblick über die einzelnen PERMEV-Komponenten und deren Schnittstellen.

Im Messsystem ‚monitor' werden mithilfe des SHELL-Skripts permon und der - je nach Betriebssystem - verfügbaren Standardmesstools die Performance-Daten gesammelt und als Rohdaten in einzelnen Dateien abgelegt. Die Konfigurationsdatei permon.cfg definiert, welche Tools aufgerufen werden. Eine Messung erfolgt über einen in Messintervalle gegliederten Messzeitraum. Am Schluss einer Messung werden diese Daten in der Archivdatei komprimiert verpackt.

Im Auswertesystem ‚evaluation' werden mithilfe des PERL-Skripts eval.pl und weiterer PERL-Moduln aus den Rohdaten über eine Vorauswertung erste Ergebnistabellen erzeugt. Anschließend können diese Ergebnistabellen durch eine Datenverknüpfung zu weiteren Ergebnistabellen kombiniert werden. Die Vorauswertung wird gesteuert durch die Konfigurationsdatei preeval.cfg. Auf Basis aller Ergebnistabellen wird abschließend eine tabellarische Ergebnisübersicht generiert, deren Umfang über die Konfigurationsdatei eval.cfg variabel spezifizierbar ist. In den Ergebnistabellen und der Ergebnisübersicht sind die einzelnen Performance-Daten zeitlich den jeweiligen Messintervallen zugeordnet.

Im Präsentationssystem ‚statistics' erfolgt mithilfe eines MS-Excel-Makros eine weitere u.a. grafische Aufbereitung der Ergebnisübersicht. Die Konfigurationsdatei statcfg.txt legt Form und Umfang der dabei resultierenden Tabellen und Diagramme fest.

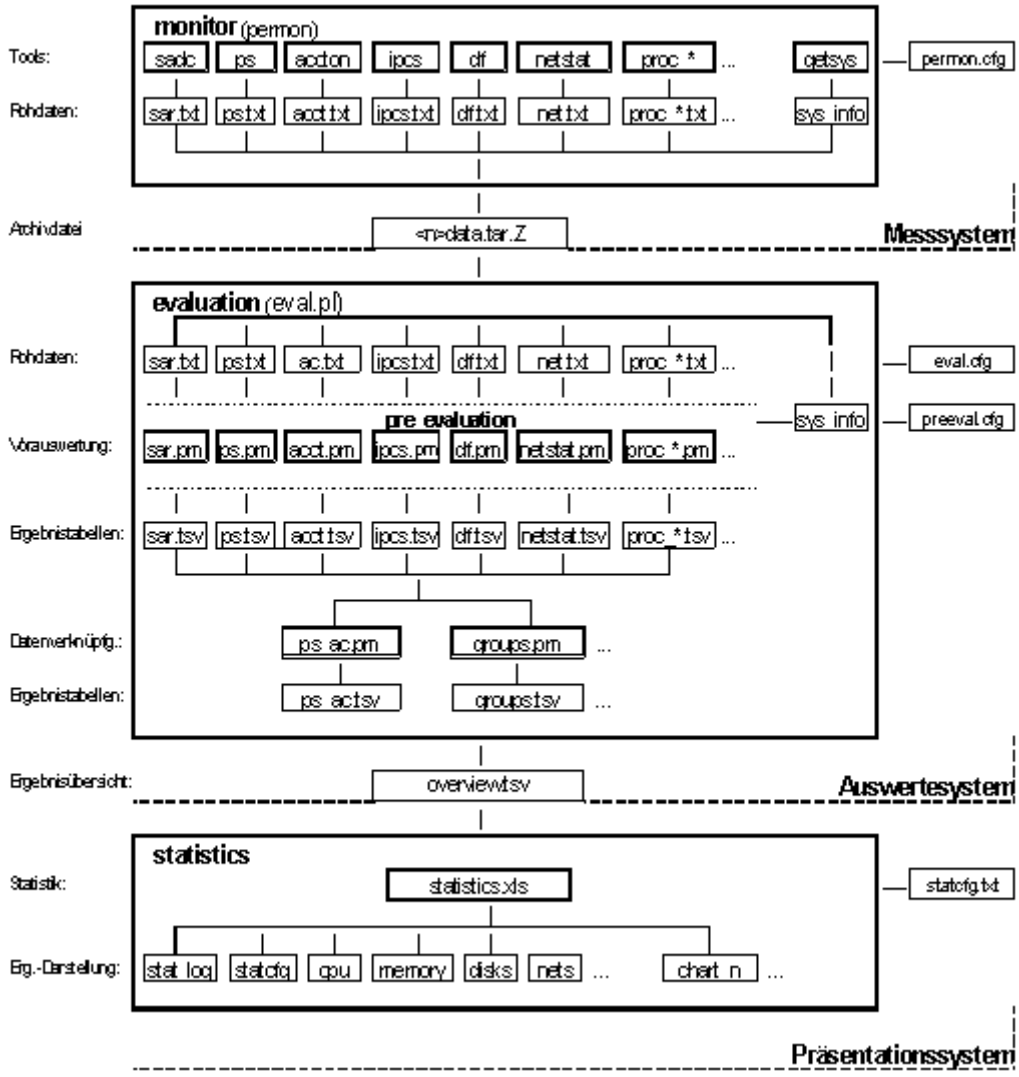


Abbildung 01-1: PERMEV-Ablauf- und -Schnittstellenstruktur
(Darstellung für eine Einzelmessung)

(b) Das Messsystem

Aufgabe

Das Messsystem ‚monitor' dient zur mittel- und langfristigen Erfassung system- und anwendungsbezogener Performance-Daten. Die Messdatensätze werden in definierbaren Abständen, d.h. den Messintervallen, mittels Standardmesstools gesammelt und in einem Ausgabefile komprimiert abgelegt. In einer Einzelmessung werden Daten von maximal einem Tag (24 Stunden, ohne Datumsüberschreitung) erfasst. Eine Langzeitmessung besteht aus einer Folge von Einzelmessungen. Die Auswertung erfolgt ausschließlich offline.

(c) Das Auswertesystem

Aufgabe

Das Auswertesystem ‚evaluation' kann - mit einigen Einschränkungen (s.u.) - unabhängig vom Betriebssystem des Messobjekts, d.h. des Systems, auf welchem die Messungen durchgeführt worden sind, unter Reliant UNIX, SOLARIS oder LINUX eingesetzt werden. Voraussetzung ist eine PERL-Installation.

Das Auswertesystem nimmt an der Schnittstelle zur Komponente ‚monitor' Rohdaten in Form einer verpackten und komprimierten Datei entgegen und verarbeitet diese zu Ergebnistabellen. In diesen Ergebnistabellen werden die verschiedenen Performance-Daten den jeweiligen Messintervallen zugeordnet. Durch eine entsprechend dem Analyseziel parametrisierbare Datenreduktion werden aus den Ergebnistabellen Performance-Daten selektiert und in einer Ergebnisübersicht abgespeichert. Die Ergebnisübersicht bildet die Grundlage für die Ergebnisdarstellung und für weitere statistische Auswertungen.

Eine Auswertung von LINUX-Messdaten unter Reliant UNIX oder SOLARIS bedarf jedoch einiger vorheriger Anpassungen. Diese betreffen das Messdaten-Archivierungsverfahren unter LINUX (s.o.), welches Reliant UNIX- bzw. SOLARIS-kompatibel im *permon*-Skript zu ändern ist, sowie das C-Konvertierungsprogramm für LINUX-a*ccounting*-Daten (s.u.), welches auf dem Reliant UNIX- bzw. SOLARIS-System anzupassen und neu zu übersetzen ist.

(d) Das Präsentationssystem

Das Präsentationssystem ‚statistics' dient dazu, auf Basis von MS-Excel die tabellarische Ergebnisübersicht grafisch aufzubereiten. Zudem können Korrelationen vorgegebener Performance-Kenngrößen berechnet und in Abhängigkeit des jeweiligen Korrelationswertes ebenfalls grafisch dargestellt werden.

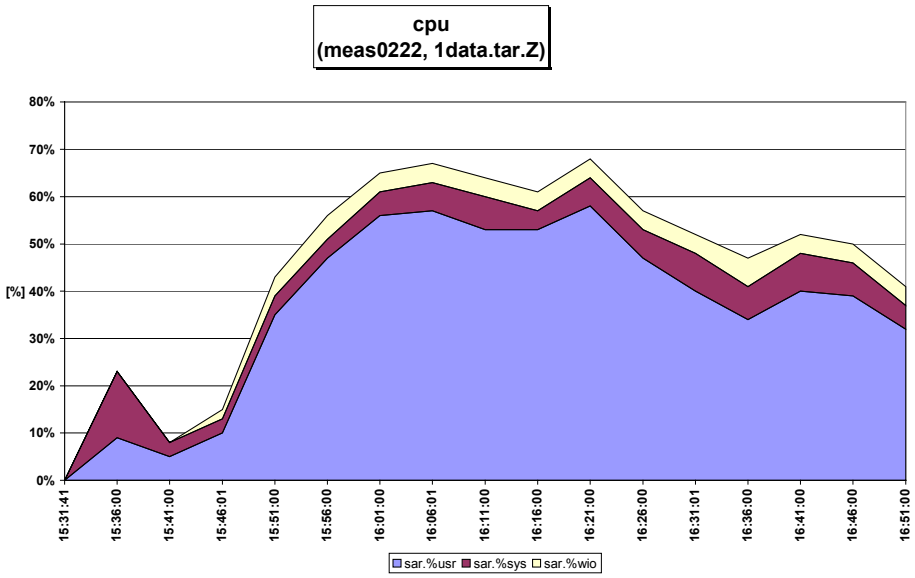


Abbildung 01-2: CPU-Auslastung, Anteile von User-Prozessen, Systemoverhead und wait-I/O

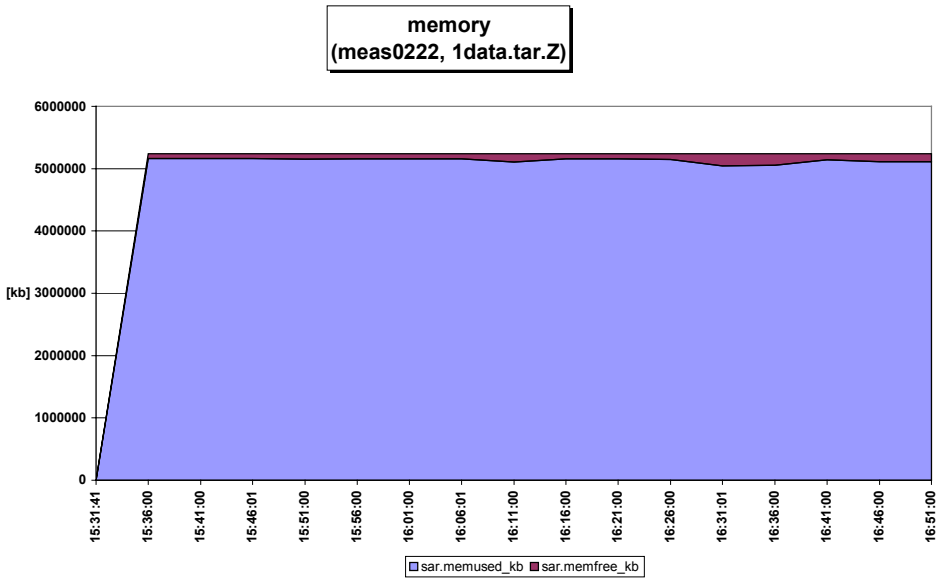


Abbildung 01-3: Speicherbelegung

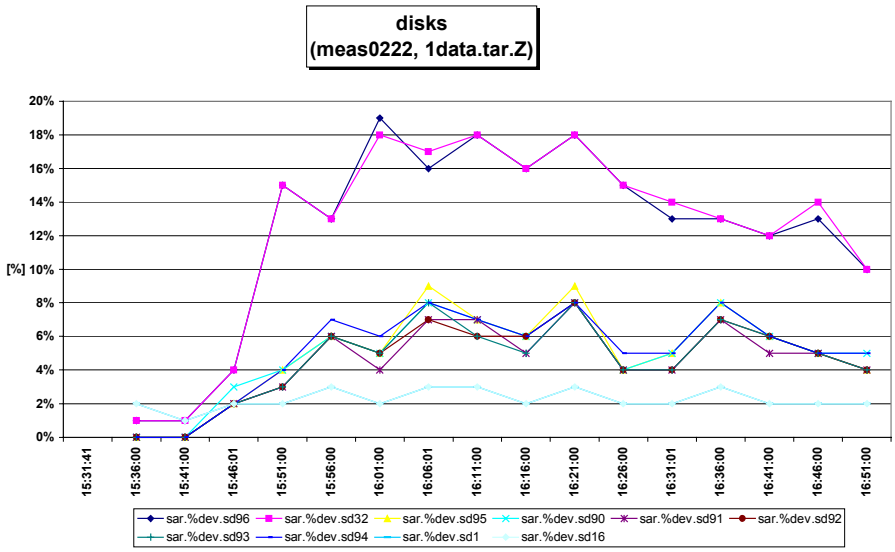


Abbildung 01-4: Plattenauslastung

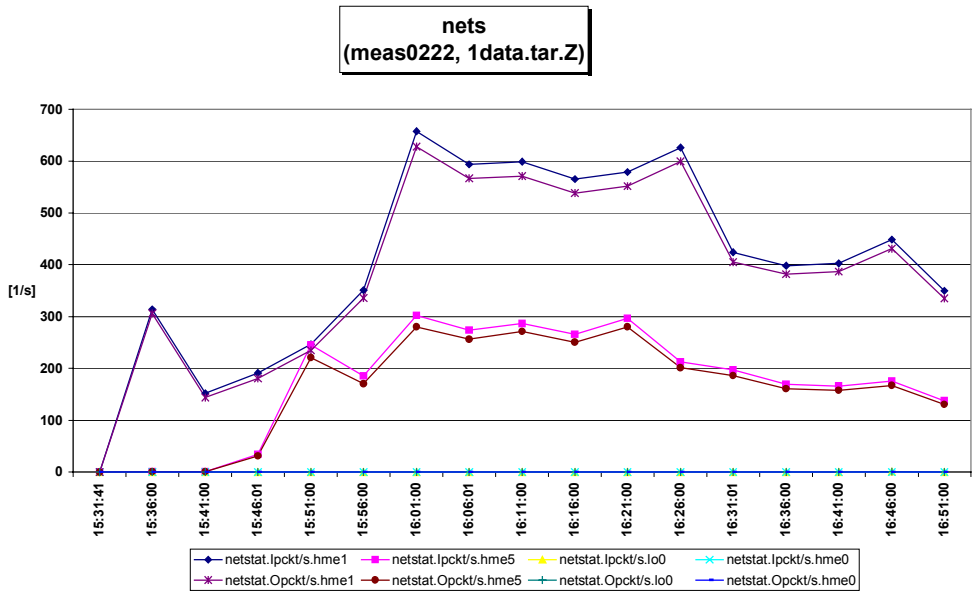


Abbildung 01-5: Netzstatistik

KAPITEL 02 PROAKTIVES KAPAZITÄTSMANAGEMENT

REINHARD BORDEWISCH, KURT JÜRGEN WARLIES

02.01 Einführung und Motivation

Die Hinwendung zur New Economy stellt die Unternehmen vor eine hohe Herausforderung. Die Geschäftsprozessketten müssen dazu durchgehend, das heißt vollständig IT-gestützt ablaufen. In dieser Form konzipiert, kommen die Geschäftsprozesse auch der Old Economy zugute. Das funktioniert allerdings nur dann, wenn alle Systeme – Netzwerk, Server, Anwendungen – im Sinne „eines" Geschäftssystems nahtlos zusammenwirken.

Nur wie dieses nahtlose Zusammenspiel effizient und sicher auf den Weg bringen, ohne schnurstracks in Design-, Integrations-, Verfügbarkeits- und Ablaufprobleme hinein zu laufen? Zumal mit der Entwicklung eines umfassenden Geschäftssystems die Systeme, Schnittstellen und Ereignisse sich gegenseitig beeinflussen. Die Folge: Anvisierte Größen wie Skalierfähigkeit, Flexibilität, Verfügbarkeit, Performance und Dienstgüte sind nicht mehr vorhersehbar und damit nicht konkret planbar.

02.02 Im Brennpunkt: Gesamtsystemverhalten aus Endbenutzersicht

Hochverfügbarkeit, Zuverlässigkeit und Dienstgüte sind wichtige Kriterien einer modernen IT-Landschaft. Wie werden diese Kriterien messbar und damit überprüfbar und abrechenbar? Wie können diese Anforderungen sichergestellt werden?

Ein bereits früher verfolgter Ansatz ist die Überwachung der Server-Ressourcen mittels der von Siemens Business Services (SBS) Paderborn entwickelten automatisierten Mess- und Auswerteumgebung PERMEV zur Server-Langzeitüberwachung. Ein inzwischen immer mehr ins Blickfeld gerückter Aspekt ist das Systemverhalten am Client, wie es sich dem Benutzer gegenüber präsentiert. Das schließt auch die Verweilzeiten im Netzwerk und im Client ein. Subjektiven Äußerungen der Anwender über ihr langsames System können IT-Betreiber nur selten etwas entgegensetzen. Heutige Netzwerkmanagement-Tools basieren auf der Ermittlung der Auslastungsgrade der System- und Netzwerk-Komponenten. Überwachungsmechanismen innerhalb der Anwendungen sind selten oder fehlen ganz. Konkret bemängelten die Endanwender einer Bundesbehörde die unbefriedigende Dienstgüte (das schlechte Antwortzeitverhalten) der Bearbeitung ihrer Geschäftsprozesse, so dass der Leiter des Rechenzentrums den objektiven Nachweis der Dienstgüte verlangte.

Dies war für Siemens Business Services (SBS) Paderborn die Motivation für die Entwicklung des Referenz-PC unter MS-Windows. In der realen Infrastruktur werden auf einem exklusiv verfügbaren Client-PC business-kritische Applikationen zyklisch zum Ablauf gebracht, wobei der Endbenutzer durch einen automatischen Testtreiber simuliert wird. Der Referenz-PC beinhaltet einen solchen „Testautomat“, der einen realen Benutzer an der grafischen Oberfläche simuliert. Business-kritische Transaktionen werden in festgelegten Abständen durchgeführt und deren Antwortzeit angelehnt an die DIN 66273 ermittelt und protokolliert. Die realen Eingaben (Tastatureingaben, Mausklicks) werden erfasst und mit konfigurierbaren Wartezeiten an die Anwendung übergeben. Während der Ausführung werden die relevanten Antwortzeiten ohne Modifikation der zu überwachenden Applikation und ohne Rückkoppelungseffekte auf das Gesamtsystem ermittelt.

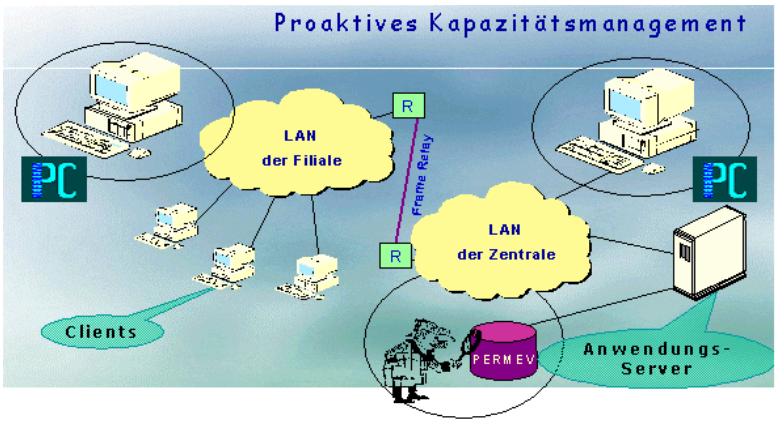


Abbildung 02-1: Gesamtsystemsicht des proaktiven Kapazitätsmanagements

Mit dem Einsatz des Referenz-PC werden die folgenden Zielsetzungen verfolgt:

- Das Antwortzeitverhalten beliebiger Applikationen in Client-/Server-Umgebungen aus Benutzersicht wird messbar und objektiv qualifizierbar.
- Nach Kalibrierung der Schwellwerte erkennt der Referenz-PC einen (drohenden) Leistungsengpass früher als der Anwender und stellt dem Administrator diese Information mittels Alerting zur Verfügung.

Trendanalysen über die vermessenen Transaktionen ermöglichen eine planerische Reaktion auf Leistungseinschränkungen.

02.03 Automatisierte Messungen mittels PERMEV

Es treten wiederkehrende Fragestellungen der IT-Betreiber auf wie:

- Das Antwortzeitverhalten ist unbefriedigend, der Transaktionsdurchsatz liegt weit unter dem tatsächlichen Lastaufkommen. Man vermutet einen CPU-Engpass, aber wie steht es mit I/O-Peripherie, Hauptspeicher und Netz?
- Reicht die Hardware-Kapazität der System-Ressourcen und Netzbandbreiten für die nächste Ausbaustufe der Anwendungsumgebung aus?

PERMEV ist eine automatisierte Mess- und Auswertungsumgebung für den Einsatz auf unterschiedlichen Unix-Server-Plattformen - Linux, Solaris, Reliant UNIX - welche die Aufrufe von Kommandos wie z.B. mpstat, ps, acct und netstat standardisiert, synchronisiert und die erhobenen Performance-Rohdaten über alle Erfassungsquellen auswertet. Das im PERMEV-Messsystem enthaltene Monitoring-Skript 'permon' wird zur mittel- und langfristigen Performance-Messung auf UNIX-Systemen eingesetzt. Für die Durchführung der automatisierten Messungen wird das komplette Messsystem wirtschaftlich von zentraler Stelle aus auf dem (den) zu vermessenden Server(n) kopiert, konfiguriert und aktiviert. Das Skript erlaubt den konfigurierbaren Einsatz der o.g. Werkzeuge und die Aufnahme von relevanten System-HW-/SW-Informationen. Es handelt sich bei den aufgenommenen Performance-Daten vor allem um Informationen über die Belegung der Systemressourcen wie CPU, Hauptspeicher, Festplatten und Netzzugang.

Darüber hinaus ist das Monitoring-Skript schnell und einfach um zusätzliche Messwerkzeuge (Traces oder Statistiken) des Betriebssystems oder auch einer speziellen Anwendung erweiterbar. So ist z.B. in PERMEV ein Mess-Skript integriert worden, mit dem Performance-relevante Daten über eine ggf. installierte ORACLE-Datenbank erhoben werden können.

Neben einer Standardauswertung mit den wesentlichen Performance-Kenngrößen des vermessenen Systems kann die Zusammensetzung dieser Ergebnisübersicht vielfältig individuell konfiguriert werden. U.a. erlaubt PERMEV die Verknüpfung verschiedener Rohdaten, so z.B. der prozessspezifischen Daten, die einerseits aus dem ps-Kommando resultieren und andererseits durch das Accounting erzeugt werden. Zusätzlich ist es möglich, Prozesse anwendungsspezifisch und damit verursachergerecht zusammenzufassen und diese Prozessgruppen hinsichtlich ihres Ressourcenverbrauchs auszuwerten.

Die so gewonnene Ergebnisübersicht ermöglicht die schnelle Performance-Analyse des Systems und die Lokalisierung möglicher Performance-Engpässe, so dass hieraus Ansatzpunkte für Tuning-Maßnahmen und/oder die Notwendigkeit für tiefergreifende Analysen abgeleitet werden können. Weiterhin können diese Performancedaten eine Basis zur Ableitung des zukünftigen Performance-Verhaltens bevorstehender Ausbaustufen oder Konfigurationsänderungen in der IT-Landschaft bilden.

Mit Hilfe des auf Basis von MS-Excel implementierten PERMEV-Präsentationssystems ist es außerdem möglich, die tabellarische Ergebnisübersicht grafisch aufzubereiten. Durch diese Visualisierung sind Performance-Engpässe sowie Spitzenbelastungszeiträume sofort erkennbar. So analysiert und in Übersicht gebracht, werden frühzeitig Performance-Engpässe und potenzielle Fehlerzustände im Server-Bereich deutlich. Das wiederum ermöglicht, Server und Verbindungen richtig zu dimensionieren, Tuning-Maßnahmen gezielt aufzusetzen, Verarbeitungstrends im Server-Bereich frühzeitig zu erkennen und Server-Farmen angemessen zu formieren.

02.04 Proaktives Kapazitätsmanagement

Heutige Netzwerkmanagement-Tools basieren i.d.R. auf der Überwachung der Auslastungsgrade der System- und Netzwerk-Komponenten und ermöglichen nur sehr eingeschränkt Detail- und Ursachenanalysen des Systemverhaltens. Ein bereits früher seitens SBS verfolgter Ansatz ist die Überwachung und Analyse der Server-Ressourcen mittels PERMEV. Ziel des Proaktiven Kapazitätsmanagements ist es, bei Verletzung der Dienstgüte gezielt Detail- und Ursachenanalysen hinsichtlich des Verhaltens der Systemkomponenten zu erstellen, um den IT- und Applikationsbetreiber in die Lage zu versetzen, frühzeitig Maßnahmen einzuleiten, bevor sich ein Anwender beschwert.

Das Proaktive Kapazitätsmanagement auf Basis der Koppelung des Referenz-PC mit der bewährten PERMEV-Langzeitüberwachung bietet zusätzliche Vorteile: Die PERMEV-Langzeitüberwachung ermittelt die Ressourcenauslastung am Server und stellt eine Trendanalyse über einen vorzugebenden Zeitraum bereit. Die Messintervalle sind dabei relativ groß gewählt, um das System nicht unnötig zu belasten.

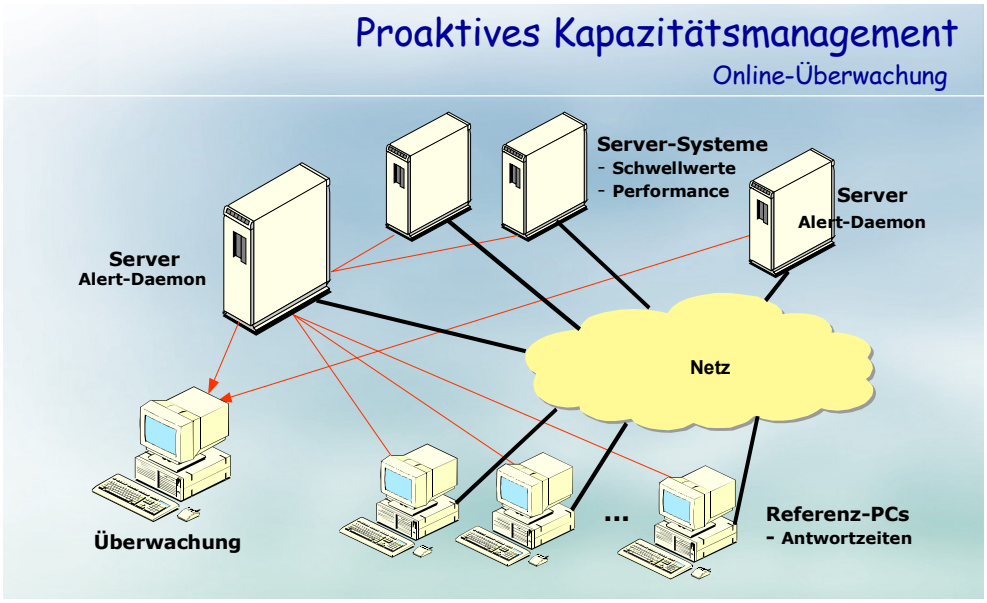


Abbildung 02-2: Online-Überwachung von Serversystemen

Bei Schwellwertüberschreitungen am Client wird neben einem Alert an den Administrator auch zusätzlich das PERMEV-Messintervall verkürzt. Man erhält so eine höhere Granularität der Messdaten. Durch die Korrelation der Messdaten mit den Antwortzeiten wird eine gezielte Ursachenanalyse des Server-Systems ermöglicht. Gleichzeitig werden für einen kurzen Zeitraum die Client-Transaktionen in verkürztem Abstand durchgeführt, um das Antwortzeitverhalten bis zu einer „Normalisierung“ detailliert zu verfolgen.

Der Nutzen für den IT- und Applikationsbetreiber liegt in der Option, frühzeitig Maßnahmen einleiten zu können, bevor sich ein Anwender beschwert. Mögliche schnelle Aktionen sind die Zuschaltung von Leistungsressourcen (Server, Netz, Applikation, verteilte Ressourcen, ...) oder Abschalten nicht zeitkritischer Hintergrundlasten.

Bei Einsatz mehrerer Referenz-PCs, z.B. in unterschiedlichen Netzsegmenten, sind Aussagen zur jeweiligen Providerleistung möglich. Die gemessenen Antwortzeiten können als Basisdaten für die Überwachung von Service-Level-Agreements herangezogen werden.

02.05 Online-Auswertung der Serverressourcen

Die Statusmeldungen des Referenz-PC werden an einer eigenen grafischen Oberfläche zentral im Operating angezeigt. Um nun jedoch parallel die Auslastungsgrade der beteiligten Server beobachten zu können, ist eine Erweiterung der PERMEV-Funktionalität notwendig. Analog zu den Daten des Referenz-PC werden Schwellwerte für die Auslastung der Serverressourcen CPU, Speicher, Platte und Netz definiert und in einer „Miniauswertung“ **online** erhoben und geprüft. Die Statusmeldungen werden zusammen mit denen des Referenz-PC angezeigt und ermöglichen so eine Gesamtbetrachtung der Applikationskomponenten.

02.06 Anwendungsgebiete

Außer in der beschriebenen Client-/Server-Applikation wurde das Proaktive Kapazitätsmanagement für eine R/3-Anwendung eines großen Halbleiterherstellers adaptiert. Der Einsatz im ASP-Umfeld wird gerade vorbereitet.

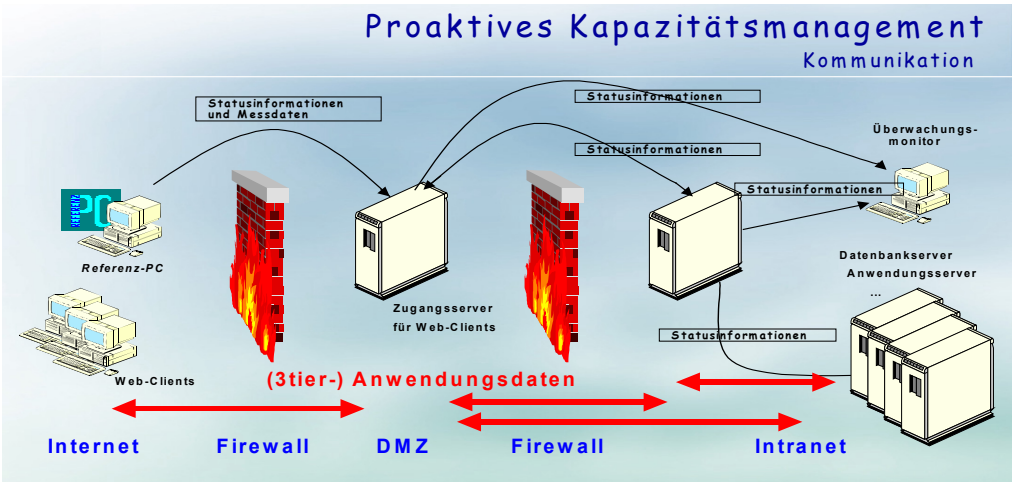


Abbildung 02-3: Kommunikation über drei Sicherheitsebenen

Hier geht es um die Überwachung von business-kritischen Transaktionen, die ein mittelständischer Kunde bei einem Applikation Service Provider (ASP) bestellt hat und die im Rahmen eines Unternehmensportals für die Mitarbeiter über einen Web-Browser bereitge-

stellt werden. Eine Besonderheit bedeutet dabei die Kommunikation der einzelnen Komponenten über die drei Sicherheitsebenen Internet, DMZ, Intranet des RZ-Betreibers.

Somit ermöglicht das Proaktive Kapazitätsmanagement ein Agieren statt Reagieren: Von der Problemerkennung zur Problemerkennung und -vermeidung!

KAPITEL 03 APPLICATION EXPERT

KLAUS HIRSCH

03.01 Übersicht

Das Tool Application Expert eignet sich gut zur Performance-Optimierung beim Betrieb von mehrstufig vernetzten Client/Server-Anwendungen. Application Expert ist ein Produkt der Firma Compuware (früher Optimal Networks). Haupteinsatzfeld ist die Analyse von verteilten Anwendungen in Bezug auf deren Laufzeitverhalten bei gleichzeitigem Aufzeigen von konkreten Tuningansätzen.

Unmittelbarer Nutzen und Erkenntnisse beim Einsatz von Application Expert lassen sich wie folgt kategorisieren:

- Nachweis der Gesamtlaufzeit eines konkreten Geschäftsprozesses aus Sicht des End-Users
- Exakte Zuordnung der Zeitanteile zu den involvierten HW-Komponenten (Client, Netz, Applikationsserver, Datenbankserver) und Erkennen zeitlicher Synchronisationsmuster bei verteilt ablaufenden Transaktionen
- Nachweis des Datenverkehrsaufkommens zwischen den beteiligten Komponenten
- Gezieltes Erkennen von Tuningbedarf und -möglichkeiten bereits vor einem produktiven Einsatz von C/S-Anwendungen aufgrund komfortabler Facilities zur Top-down-Analyse
- Prognose zum Laufzeitverhalten von Geschäftsprozessen bei variierender Netzbandbreite
- Automatisierte, grafische und tabellarische Aufbereitung von Ergebnissen

Application Expert ist auf allen Windows-Plattformen einsetzbar. Die überwachten HW-Objekte können von beliebiger Art sein. Die Erfassung der Messdaten erfolgt über das Netzwerk-Interface eines PCs oder Notebooks oder durch den Import von Trace-Files von Protokolltestern oder Netzwerkmanagement-Tools. Soweit grundlegende Kenntnisse im Bereich Performance und Network-Monitoring vorliegen, ist der effiziente Umgang mit Application Expert relativ leicht zu erlernen.

03.02 Instrumentierung und Analysemethoden

Die Instrumentierung des Application Expert kann man für konkrete Untersuchungszwecke unterschiedlich anpassen. Durch entsprechendes Setzen von Filtern lässt sich gezielt der Datenverkehr zwischen zwei einzelnen IP-Adressen, mehreren Paaren von IP-Adressen oder auch zwischen einer IP-Adresse und allen anderen Adressen, die mit dieser Daten austauschen, mitschneiden. Es ist aber auch möglich, den gesamten Datenverkehr zu erfassen. Je nach Art der im Monitorsystem eingebauten Netzwerkkarte ist es so möglich im Ethernet, Fast-Ethernet, FDDI oder Token Ring zu messen.

Durch spezielle Importfeatures ist es möglich Messungen von Protokolltestern (z.B. Wandel und Goltermann) oder Netzwerkmanagementsystemen (z.B. HP Openview) zu analysieren.

Neben den umgehend verfügbaren Standardergebnissen erlauben es die oben dargestellten Diagnosemöglichkeiten somit, das Laufzeitverhalten von Geschäftsprozessen Top-down zu untersuchen und ggf. einzelne Aufrufe auf Applikationsebene als besonders ressourcen- und zeitintensiv zu identifizieren.

Darüber hinaus ist es möglich, mit den Funktionen

- *Response Time Predictor* das Laufzeitverhalten einer untersuchten Transaktion bei verschiedenen Netzbandbreiten analytisch zu berechnen.
- *Bandwidth Estimator* die zwischen 2 IP-Adressen verfügbare Netzbandbreite experimentell zu ermitteln.
- *Latency Finder* die Signallaufzeit zwischen 2 IP-Adressen zu bestimmen.
- *Comparison Report* identische untersuchte Abläufe/Transaktionen automatisch zu vergleichen.

03.03 Beispiel: Mehrstufige Client/Server-Kommunikation

Nachfolgendes Beispiel zeigt die für einen Geschäftsprozess anfallenden Kommunikationsvorgänge zwischen einem Client, Applikationserver und Datenbankserver im zeitlichen Verlauf. Im Kasten rechts erkennt man den Inhalt eines einzelnen Datenpakets, während im Kasten unten der Inhalt dieses Datenpakets auf das HTTP-Protokoll reduziert wird. Neben dem HTTP-Protokoll können noch eine Reihe weiterer Protokolle, wie z.B SQL, FTP, SMTP u.a. analog decodiert werden.

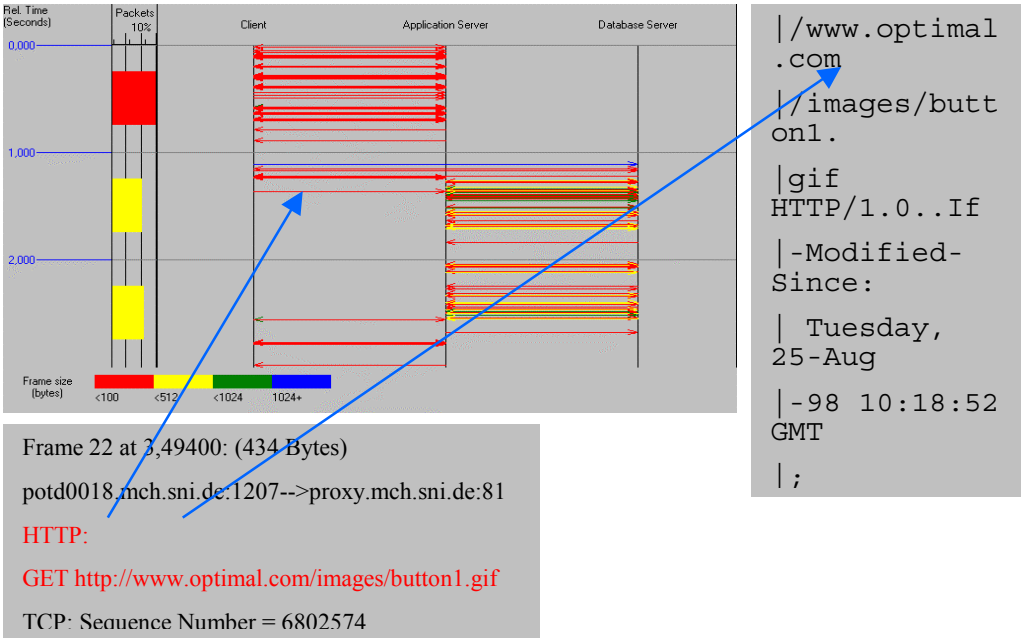


Abbildung 03-1: Application Expert im Einsatz

03.04 Abgrenzung zu anderen Messverfahren

Der Einsatzschwerpunkt des Application Expert ist weniger das Aufzeichnen des gesamten im Multiuser-Betrieb von C/S-Applikationen erzeugten Netzverkehrs, sondern vielmehr der Mitschnitt und die Analyse in Bezug auf einzelne, definierte Geschäftsprozesse.

Idealerweise sollten die mit Application Expert in einer C/S-Umgebung überwachten Objekte innerhalb eines Netzsegmentes liegen. Application Expert protokolliert den Datenverkehr zwischen den beim Ablauf eines Geschäftsprozesses beteiligten Komponenten. An den überwachten Objekten - Hardware und Software - sind aufgrund dieser Art der Messdatenerfassung keinerlei Anpassungen vorzunehmen. Es ergeben sich somit auch keine durch Mess-Overhead verursachten Rückkoppelungseffekte auf das beobachtete Ablaufverhalten.

Allerdings liefert alleine diese Art des Monitoring keine Erkenntnisse über den aktuellen Auslastungszustand der beobachteten Systeme (z.B. Clients oder Server). Man stellt aber

die direkte Auswirkung der auf diesen Komponenten verfügbaren Systemleistung auf das Laufzeitverhalten fest. Je nach Untersuchungsziel ist es dann auch möglich, die Ergebnisse mit Messdaten zu korrellieren, die zeitgleich über andere Schnittstellen (z.B. in einem Server) erfasst wurden.

Application Expert hat sich praktischen Anwendungen bestens bewährt.

KAPITEL 04 SERVICE LEVEL MANAGEMENT

STEFAN SCHMUTZ, NORBERT SCHERNER

04.01 Einleitung

Es wurden die Service Level Management Software Produkte folgender Firmen untersucht, verglichen und bewertet. Zu diesem Zweck wurden Erfahrungen beim praktischen Einsatz sowie Herstellerangaben herangezogen.

@ AMDAHL	EnView
@ BMC	SiteAngel
@ Candle	eBA*ServiceMonitor
@ HP	OpenView Vantage Points Internet Services
@ Keynote	Perspective
@ Mercury Interactive	Topaz
@ Service Metrics	WebPoint
@ Tivoli	Web Service Manager und Web Service Analyser

04.02 Fazit

Betreiber von Online-Anwendungen, die schon seit jeher ihr Geschäft mit dem Testen von Client/Server-Anwendungen machen, haben mit einer Fülle an Lösungen und Dienstleistungen reagiert. Da die Handhabung der Produkte wegen ihrer teils komplexen Technik einen hohen Einarbeitungsaufwand erfordert, wird Service Level Management häufig als Service angeboten. Daher gehen die Firmen vermehrt dazu über, ihre Programme nicht zu verkaufen, sondern deren Einsatz und das zugehörige Consulting zu vermieten, um auch kleineren Unternehmen ohne große Personalressourcen diese komplizierte Aufgabe abzunehmen. Die als Online-Service angebotenen Performance-Messungen liefern oft nur generelle Hinweise auf Schwachstellen. Erst die detaillierte Analyse der gesamten Infrastruktur gibt Auskunft darüber, wo genau nachgebessert werden muss. Wirklich neu ist von den betrachteten Lösungen nur der Ansatz von Candle. Um den Wert der angebotenen Lösungen beurteilen zu können, ist im Einzelfall die prototypische Umsetzung und der Nachweis der Praxistauglichkeit durch die Hersteller einzufordern.

04.03 AMDAHL EnView

EnView ist eine aus mehreren Komponenten bestehende Service Level Management Lösung. Sie ist skript-gestützt, d.h. die Simulation relevanter Transaktionen wird über Skripte gesteuert. Dazu setzt EnView auf dem Tool PerformanceStudio der Firma Rational auf. EnView überwacht sowohl die Verfügbarkeit als auch die Antwortzeiten beliebiger Client/Server-Anwendungen. Die Skripte können Abläufe sowohl an der Oberfläche als auch auf der Ebene des Netzwerk-Protokolls wiedergeben.

Das Produkt EnView besteht aus den Komponenten Robot, Collector, Monitor und Reporter. Die Robots werden an den Benutzerstandorten installiert und deren Ergebnisse in Echtzeit an den Collector übertragen. Der Collector sammelt die Ergebnisdaten, fasst sie zusammen und übergibt sie zur Statusanzeige an eine oder mehrere Monitor Stationen, vgl. Abbildung 04-1. Jeder Monitor liefert den Echtzeit-Status aller überwachten Anwendungen bzw. Prozesse. Zur Performance-Analyse und zur Prognose für Planungszwecke werden die Daten in einem Service Level Repository – einer MS SQL Server Datenbank gespeichert.

Mit EnView Commerce bietet AMDAHL auch einen Level Management Service an, so dass sich der Kunde nicht in die komplexe Technik der Anwendung des Produktes PerformanceStudio einarbeiten muss.

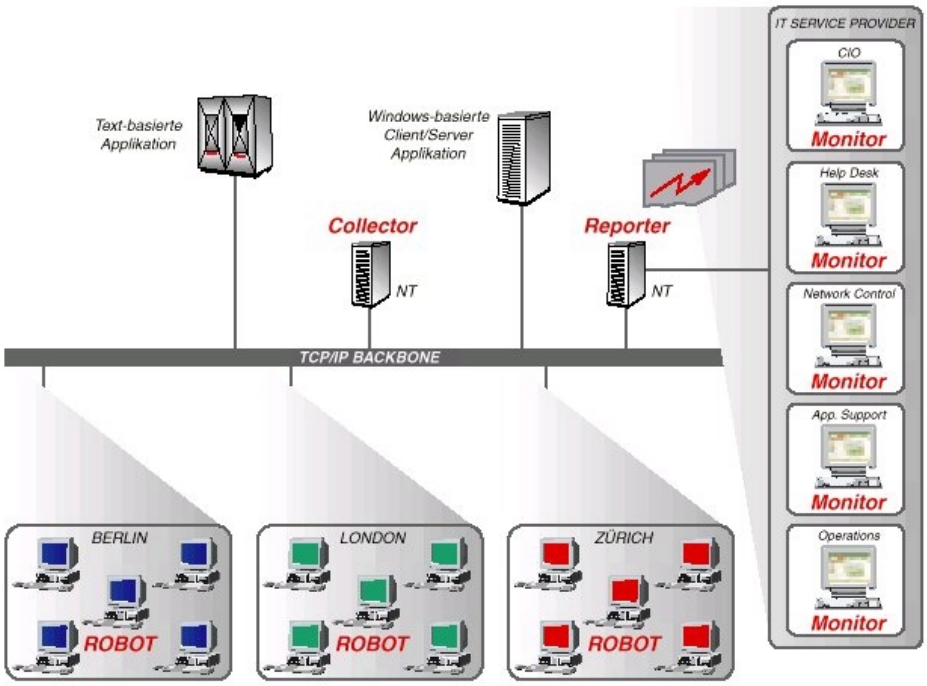


Abbildung 04-1: Komponenten von EnView

Stärken & Schwächen von EnView:

++	Unterstützung beliebiger Client/Server-Applikationen
+	Antwortzeiten und Verfügbarkeit aus End-User-Sicht
+	Reporting GUI-unterstützt
-	komplexe Technik unter Einbeziehung von Fremdprodukten
-	simulierte Transaktionen spiegeln nur bedingt reales Benutzerverhalten wider
- -	nur generelle Hinweise über mögliche Schwachstellen

04.04 BMC SiteAngel

SiteAngel verfolgt einen Ansatz, bei dem den Kunden ein Service zur Verfügung gestellt wird. Der Service kann auch bei einem durch BMC zertifizierten Partner erfolgen. Auf Client Seite ist beim Kunden keinerlei Software Installation erforderlich. Der SiteAngel Recorder läuft auf dem Web-Server bei BMC bzw. einem Partner und leitet die Requests

an den Ziel-Web-Server weiter, vgl. Abbildung 04-2. Die Anfragen und Antworten werden aufgezeichnet (sog. Teaching). Diese Aufzeichnung, Critical Path genannt, kann dann von SiteAngel zu gewünschten Zeitpunkten zum Ablauf gebracht werden (Playback). Vor dem Playback besteht die Möglichkeit, die Aufzeichnung zu prüfen (Critical Path Review) und gewünschte Verfikationen, also den Service Level, sowie Reaktionen auf Unterschreitungen und Fehler zu definieren. Neben den Antwortzeiten wird auch das Browserverhalten des Benutzers protokolliert. Die gesammelten Daten werden in einer Datenbank abgelegt, auf die mit den unterschiedlichsten Reporting Tools browser-unterstützt zugegriffen werden kann.

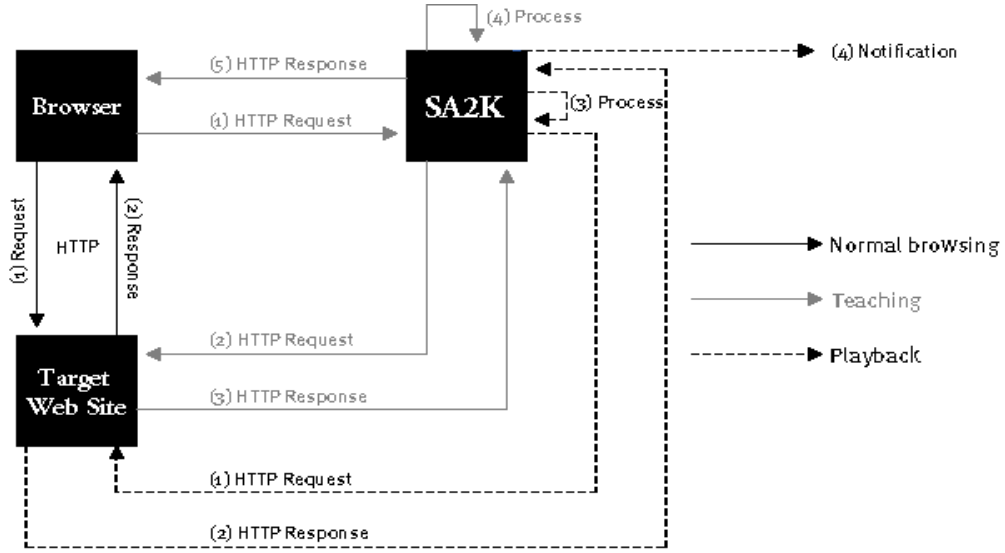


Abbildung 04-2: Schematischer Ablauf beim Einsatz von SiteAngel

Stärken & Schwächen von SiteAngel:

++ keine Client Installation
\+ keine Programmierung
\+ Web-gestütztes Reporting
\- nur für Web Applikationen
\- simulierte Transaktionen spiegeln nur bedingt reales Benutzerverhalten wider

04.05 Candle eBA*ServiceMonitor

Der Ansatz bei der Service Level Überwachung beim eBA*ServiceMonitor unterscheidet sich vollständig von dem der anderen Hersteller. Die Daten werden mit Hilfe eines Applets gesammelt, das zum Client übertragen wird und sämtliche Messwerte aufzeichnet. Auf Client Seite ist keinerlei Software Installation erforderlich. Der eBA*ServiceMonitor läuft auf dem Web-Server und misst End-to-End-Antwortzeiten aus der Sicht des Anwenders, vgl. Abbildung 04-3. Auf diese Weise wird, anders als bei den weiteren hier betrachteten Werkzeugen, beim eBA*ServiceMonitor reales und nicht simuliertes Benutzerverhalten beobachtet.

Die Antwortzeit wird vom Mausklick bis zum vollständigen Aufbau des Bildes gemessen. Neben den Antwortzeiten wird auch das Browserverhalten des Benutzers protokolliert. Die gesammelten Daten werden in Dateien oder ODBC-Datenbanken abgelegt, auf die mit den unterschiedlichsten Reporting Tools zugegriffen werden kann.

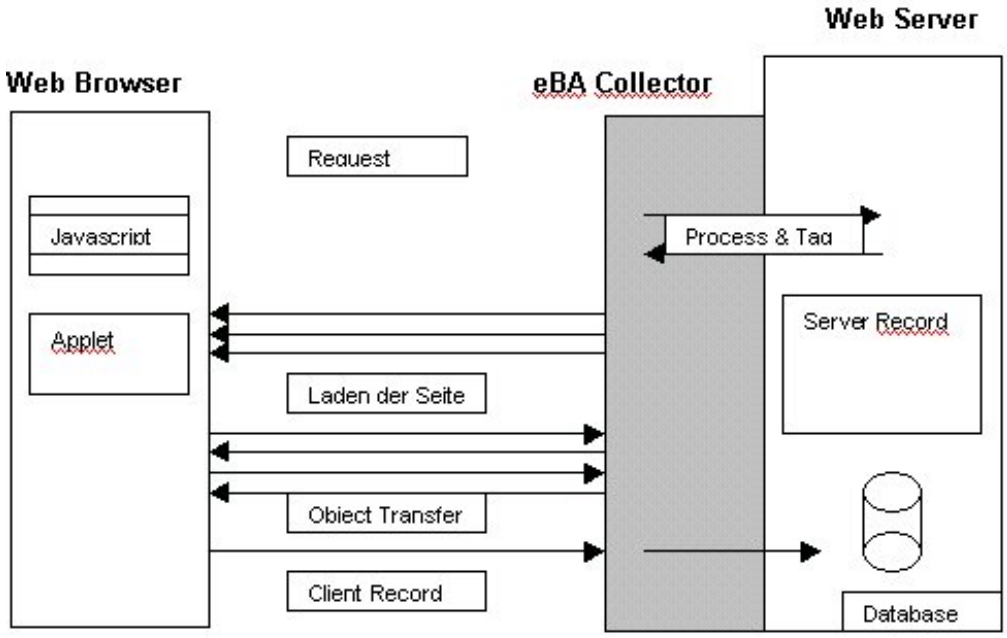


Abbildung 04-3: Einbindung eBA Collector im eBA*ServiceMonitor

Stärken & Schwächen von eBA*ServiceMonitor:

++ keine Client Installation
+ keine Programmierung
- ausgefeiltes Reporting nur über third party tools (Fa. SAS)
- nur bei Web Applikationen entfällt die Beschreibung von „Transaktions“
- - Transfer eines Applets, das Daten auf Kundenrechnern sammelt, ist problematisch

04.06 HP OpenView Vantage Points Internet Services

HP OpenView Vantage Points Internet Services ist ein Produkt aus der HP OpenView Suite zur Überwachung von Service Levels. Zum Messen der Performance werden an verschiedenen Standorten Agenten installiert. So ist es möglich die Performance aus der Sicht des Endanwenders zu messen, ohne auf dem Client des Endanwenders spezielle Software oder ein Java Applet zu laden. An Protokollen werden die wichtigsten Internet-Services unterstützt: HTTP, Secure HTTP, DNS, FTP, SMTP, POP3, IMAP, NNTP, WAP, RADIUS, LDAP und NTP (siehe Abschnitt 04.11 ‚Abkürzungen‘). Darüber hinaus kann überprüft werden, ob Rechner über ICMP erreichbar sind, Verbindungsaufbau zu TCP Ports möglich ist und Rechner über einen Dial-Up Service angesprochen werden können. Transaktionen mit mehreren HTTP-Requests sind möglich, allerdings sind die Eingaben statisch und alle Requests werden nacheinander ohne Verzögerung durchgeführt. Der Inhalt der zurückgelieferten Seiten kann automatisch verifiziert werden. Dieses Produkt ermöglicht für das HTTP Protokoll eine Kontrolle, ob der Dienst korrekt arbeitet, eine reale Nachbildung von Benutzerverhalten ist nicht möglich.

HP OpenView Vantage Points Internet Services definiert auf oberster Ebene Kunden (Customer), für die bestimmte Services überprüft werden. Für jeden Service (wie zum Beispiel HTTP, FTP etc.) können mehrere Targets angegeben werden. In der Auswertung der Messergebnisse wird Folgendes aufgeführt:

@ Targets: Jede einzelne zu überwachende Webseite oder Verbindung.

@ Services: Die Zusammenfassung mehrerer einzelner Webseiten oder Verbindungen zu einer Aussage über die Verfügbarkeit und Performance des gesamten Services. Hier kann zum Beispiel überprüft werden, wie bei mehreren überwachten Mailservern die Verfügbarkeit und Performance insgesamt war.

@ Kunden: Übersicht über die Situation für alle überwachten Services eines bestimmten Kunden.

Die Konfiguration von HP OpenView Vantage Points Internet Services erfolgt über ein übersichtliches Tool unter Windows, vgl. Abbildung 04-4:

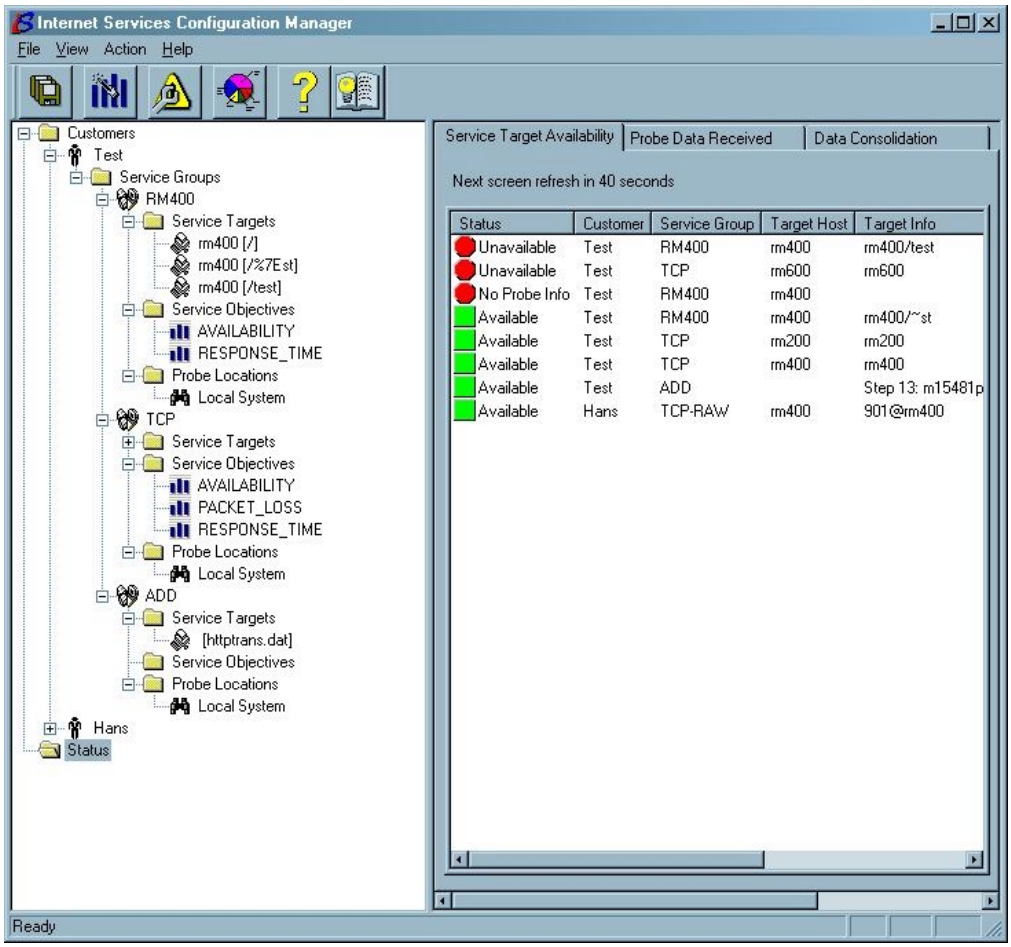


Abbildung 04-4: Oberfläche von HP OpenView Vantage Points Internet Services

Die Daten werden zentral gesammelt und gespeichert. Die Auswertung der Messdaten wird über einen Webbrowser abgefragt, hierfür ist HP OpenView Vantage Points Internet Services auf einen Internet Information Server angewiesen.

Die Daten werden in Access-Dateien gehalten, können aber auch in einer Oracle Datenbank gespeichert werden. Der Export zu anderen Tools sollte sich somit relativ einfach realisieren lassen, zumal die ODBC-Schnittstelle für den Zugriff auf die Access-Dateien bereits bei der Installation konfiguriert wird.

Stärken & Schwächen von HP OpenView Vantage Points Internet Services:

++	Unterstützt alle gängigen Internet-Dienste und kann auch beliebige TCP/IP-Ports überwachen
++	Antwortzeiten und Verfügbarkeit aus End-User-Sicht
+	Reporting Web Browser unterstützt
+	Datenaustausch über ODBC
+	keine spezielle Software oder Applets auf dem Client des End-Users
-	simulierte Transaktionen spiegeln kaum reales Benutzerverhalten sondern prüfen einzig die Verfügbarkeit des Dienstes

04.07 Keynote Perspective

Keynote's Perspective ist ausschließlich mit einem Service für das Service Level Management vertreten. Es stellt dem Kunden einen eigenen browser-gestützten Recorder zur Verfügung, der nicht auf einer Skript-Sprache basiert. Die Aufnahme des Recorders muss dann an Keynote geliefert werden, welche die Aufzeichnung von sogenannten Agenten Rechnern an Keynote Standorten in gewünschter Häufigkeit abspielen. Dort werden auch Messwerte wie Antwortzeiten, DNS-Lookup-Zeiten und weitere gesammelt und archiviert. Ferner umfasst der Service Pager Alarme, Email und ein Diagnose Zentrum.

Stärken & Schwächen von Keynote Perspective:

+	außer der Aufzeichnung keine Aufwände des Kunden
-	schmales Spektrum an Performance Informationen
-	simulierte Transaktionen spiegeln nur bedingt reales Benutzerverhalten wider
- -	für Applikationen mit variablen Seiteninhalten und Requests praktisch ungeeignet

04.08 Mercury Interactive Topaz

Topaz ist wie EnView eine skript-gestützte Service Level Management Lösung. Dazu setzt Topaz auf dem Tool LoadRunner auf, das selbst zum Produktspektrum von Mercury Interactive gehört, vgl. Abbildung 04-5. Topaz überwacht sowohl die Verfügbarkeit als auch die Antwortzeiten beliebiger Client/Server-Anwendungen. Zum Sammeln der Performancedaten werden sogenannte Agenten eingesetzt, die programmierbar sind und auf dem Host laufen. Der Zugriff auf die ermittelten Daten ist über einen Web Browser möglich. Dabei kann über verschiedene Detailierungslevel die Ursache für vorhandene Probleme eingegrenzt werden.

Mit ActiveWatch und ActiveTest bietet Mercury Interactive auch einen Level Management Service und einen Test Service an, so dass sich der Kunde auch hier nicht in die komplexe Technik der Anwendung der Produkte Load-/WinRunner einarbeiten muss.

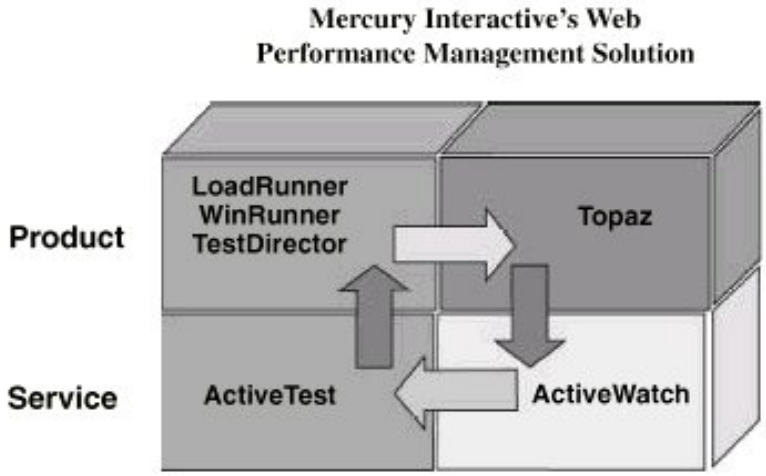


Abbildung 04-5: Produkt-/Service-Spektrum von Mercury Interactive

Stärken & Schwächen von Mercury Interactive Topaz:

++ Unterstützung beliebiger Client/Server-Applikationen
\+ Antwortzeiten und Verfügbarkeit aus End-User-Sicht
\+ Reporting Web Browser unterstützt
\- komplexe Technik
\- simulierte Transaktionen spiegeln nur bedingt reales Benutzerverhalten wider

04.09 Service Metrics WebPoint

Service Metrics WebPoint bietet sowohl Stand-Alone-Software als auch einen Service zum Service Level Management an. Die Software besteht aus einem Agenten auf dem Kunden PC, der nach dem Start für jeden Mausklick auf dem PC Performancedaten zum vollständigen Aufbau der gewählten Seite in graphischer Form darstellt, vgl. Abbildung 04-6. Dabei gliedert die Auswertung die Zeiten nach Übertragungs-, DNS-Auflösungs-, Server- und Verbindungsaufbau-Zeit für die diversen Komponenten wie Images, Binär-Daten, Java-Skripts und anderes mehr.

Als Besonderheit wird ein detailliert konfigurierbares Service Level Agreement Konzept mit Alarmstufen und Eskalations- bzw. De-Eskalations-Listen geboten. Gemäß den definierten SL werden unterschiedliche Gruppen benachrichtigt, wenn sich die Performance verschlechtert oder verbessert.

Der Miet-Service von Service Metrics umfasst neben der Administration und der Beobachtung der Web-Performance ein Consulting Angebot für die Analyse und Diagnose der ermittelten Antwortzeiten.

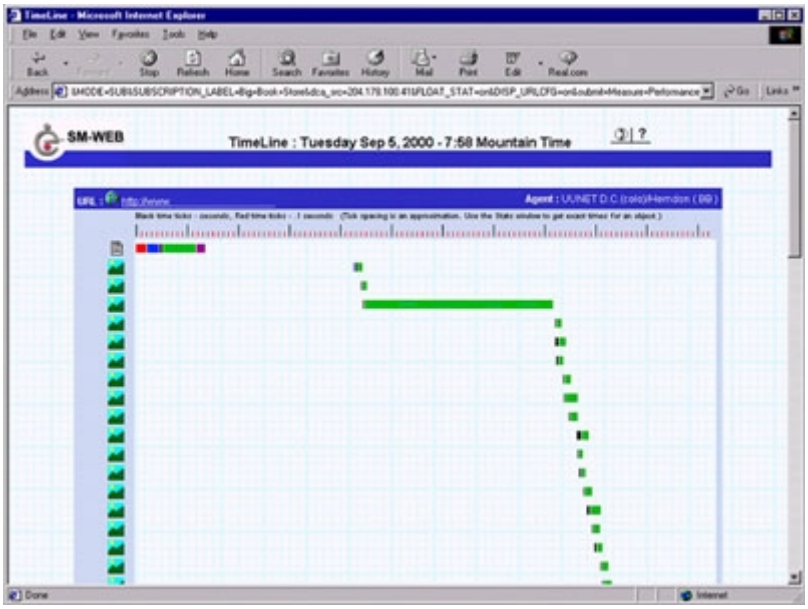


Abbildung 04-6: Darstellung der Performance Daten in WebPoint

Stärken & Schwächen von Service Metrics WebPoint:

++ differenzierte konfigurierbare und anpassbare Komponenten
+ ausgefeiltes graphisches Reporting
- überwacht nur Web Applikationen

04.10 Tivoli Web Service Manager und Web Service Analyser

Der Tivoli Web Server Manager misst die Response-Zeiten aus der Sicht des End-Users, ohne dass auf dem Client Software installiert oder ein Applet geladen wird. Für die Messungen werden vorher mit dem Tivoli Web Service Manager Transaction Investigator auf-

gezeichnete Transaktionen verwendet. Bei Überschreitung vorher festgelegter Grenzwerte kann eine e-Mail verschickt, ein SNMP-Trap generiert oder eine Meldung an die Tivoli Enterprise Konsole geschickt werden. Dabei ist es möglich, die vom Server ausgelieferten Seiten nach fehlerhaften Links und Inhalt (zum Beispiel Copyright, veraltete Produktbezeichnungen, Firmenlogo etc.) zu überprüfen.

Mit dem Tivoli Web Service Analyser können die mit dem Web Service Manager ermittelten Daten mit den Logfiles der Web- und Applikation-Server zusammengefasst werden. Alle Daten werden zentral gesammelt und können mit dem Tivoli Decision Support ausgewertet werden. Die Übertragung der Messdaten und Logfiles erfolgt über die Protokolle HTTP oder HTTPS. Dies ist wichtig, falls zwischen den verschiedenen Standorten Firewalls und Proxy-Server im Einsatz sind. Durch die Kombination der Logfiles mit den Performance-Daten ist es möglich, das Verhalten der Kunden im Verhältnis zur Verfügbarkeit und Performance der Application zu betrachten. Die Messdaten werden in einer relationalen Datenbank (Oracle oder DB2) gespeichert, so dass eine Übernahme der Daten in andere Applicationen relativ einfach implementiert werden kann. Eine Trendanalyse zur Kapazitätsplanung ist in dem Tool verfügbar.

Sollten die Produkte von Tivoli zum Einsatz kommen ist es ratsam, sowohl den Web Service Manager als auch den Web Service Analyser einzusetzen, da diese eng voneinander abhängen und sich in der Funktionalität gut ergänzen. Einen Service, die Verfügbarkeit von Websites online zu prüfen, bietet Tivoli derzeit nicht an.

Stärken & Schwächen von Tivoli Web Service Manager & Web Service Analyser:

++	Antwortzeiten und Verfügbarkeit aus End-User-Sicht
+	Reporting Web Browser unterstützt
+	Verwendung einer Standard-Datenbank und HTTP/JDBC zum Datenaustausch
-	simulierte Transaktionen spiegeln nur bedingt reales Benutzerverhalten wider
--	zwei Produkte, die eigentlich nur zusammen einsetzbar sind

04.11 Abkürzungen

DNS	Domain Name System
FTP	File Transfer Protokoll
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IMAP	Internet Message Access Protocol
LDAP	Lightweight Directory Access Protocol
NNTP	Network News Transfer Protocol

NTP	Network Time Protocol
POP3	Post Office Protocol Version 3
RADIUS	Remote Authentication Dial-In User Service
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
WAP	Wireless Application Protocol

04.12 Links

http://www.amdahl.com/de/press/2000/120900.html

http://www.keynote.com/services/html/product_lib.html

http://www.exodus.net/managed_services/metrics/performance/sm_webpoint/

http://www.candle.com/solutions_t/enduser_solutions/site_performance_analysis_external/index.html

http://www-svca.mercuryinteractive.com/products/topaz/

http://www.tivoli.com/products/solutions/web/

http://www.bmc.com/siteangel/index.html

http://www.openview.hp.com/products/vpis/

KAPITEL 05 UNIVERSELLER LASTGENERATOR *SYLAGEN*

REINHARD BORDEWISCH, BÄRBEL SCHWÄRMER

05.01 Einführung

Die Komplexität vernetzter Systeme nimmt ständig zu. Es treten hier immer wiederkehrende Fragestellungen der IT-Betreiber und -Anwender auf wie:

@ Welche Auswirkungen haben Hardware- oder Software-Umstellungen auf die Funktionalität, die Verfügbarkeit und die Performance des Gesamtsystems?

@ Reicht die Hardware-Kapazität der System-Ressourcen und Netzbandbreiten für die nächste Ausbaustufe der Anwendungsumgebung aus?

05.02 Lösungskonzept *SyLaGen*

SyLaGen (Synthetischer Lastgenerator) bietet eine voll automatisierte Steuerungs- und Lastgenerierungsumgebung für die oben genannten Anforderungen. Das Werkzeug erlaubt die Spezifizierung heterogener Lastprofile über eine einheitliche Schnittstelle für alle Anwendungsfälle. Die Erzeugung verschieden gearteter Anwenderlasten, wie z.B. File-I/O, TCP/IP-Kommunikation, HTTP-Kommunikation oder auch eine aufgezeichnete reale Kundenanwendung, wird über entsprechende Module realisiert.

SyLaGen besteht aus einer beliebigen Anzahl von Client-Agenten als Lastgeneratoren, einem oder mehreren Server-Agenten und einem Master. Die Lastgeneratoren laufen im Rahmen einer bereits installierten und konfigurierten Systemumgebung auf den Client-Systemen ab und bringen so eine definierte, reproduzierbare Last auf das Netz und die involvierten Server-Systeme.

Dabei können auch Konfigurationen simuliert werden, deren Hardware nicht in vollem Umfang zur Verfügung steht. Alle Testläufe bzw. Messungen werden zentral vom Master gestartet und anschließend von diesem bedienerlos gesteuert, synchronisiert, überwacht und ausgewertet. Ein Server-Agent sammelt Server-spezifische Performance-Daten.

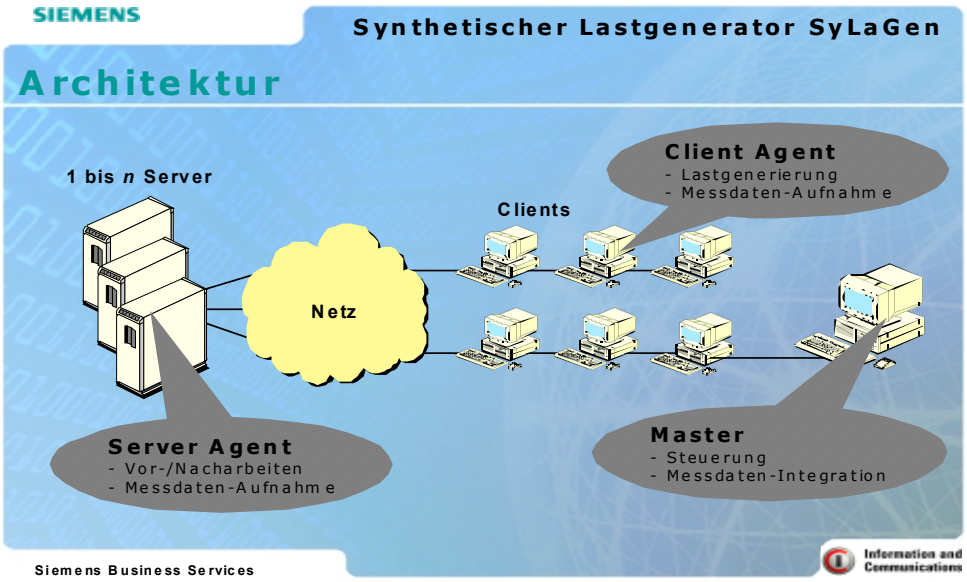


Abbildung 05-1: Architektur von *SyLaGen*

Der synthetische Lastgenerator besteht aus dem *SyLaGen* Framework und verschiedenen Lastadaptoren.

05.03 SyLaGen Framework

Die Aufgaben des *SyLaGen* Framework sind:

- @ Steuerung mit der Zustandsüberwachung des Gesamtsystems und Exploration des Lastraums
- @ Synchronisation der Systemuhren und der Phasen Vorbereitung und Durchführung der Messung sowie das Reporting
- @ Nutzung von Adaptoren, Kommunikationsadaptern für Steuerungsaufgaben und die Lastadaptoren zur Lastgenerierung
- @ Restart- und Error-Handling
- @ Reporting, wie das Fortschreiben von Log-Dateien und die Dokumentation der Messumgebung
- @ Dokumentation und Präsentation der Ergebnisse: Generierung der Ergebnisreports und grafische Darstellung der Ergebnisse „on the fly“

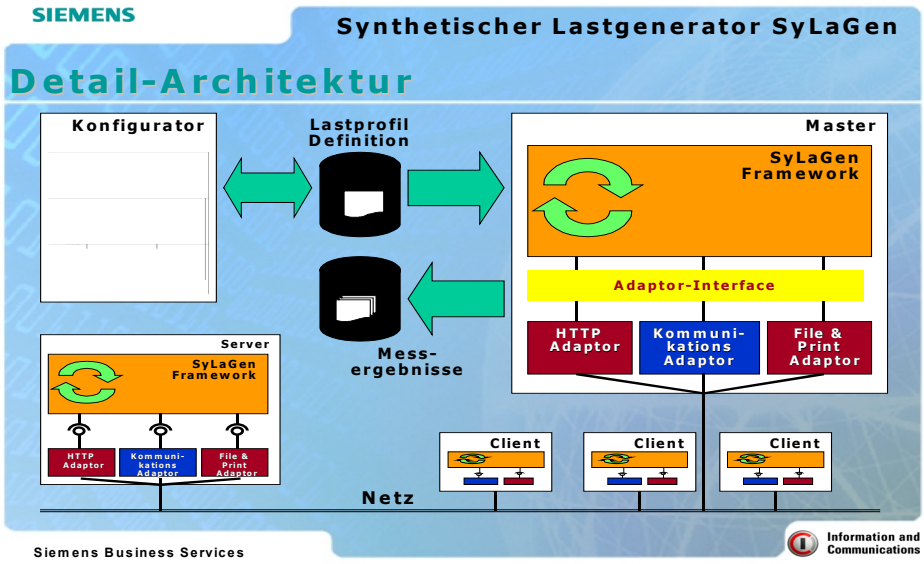


Abbildung 05-2: Detail-Architektur

05.04 SyLaGen Lastadaptor

Der *SyLaGen* Lastadaptor

- ist die Realisierung der konkreten Ausprägung eines Lastgenerators,
- dient zur Lastgenerierung durch Ausführen von Elementaroperationen, wie z.B. open (), chdir (), unlink (), read () bei File-I/O oder send (), recv () bei TCP/IP,
- existiert in verschiedenen Realisierungen und bietet zur Lastgenerierung einen (möglichst) kompletten Satz von Elementaroperationen an,
- bildet auf dem Server das Gegenstück zum die Last generierenden Client und setzt die Requests des Clients in server-lokale Ressourcenverbräuche um.

Mit *SyLaGen* können Tests bzw. Messungen sowohl mit fest vorgegebener Anzahl von Clients (Lastgeneratoren) als auch - in Form eines Benchmarklaufs - mit einer automatisch variierenden Anzahl von Clients durchgeführt werden. Bei einem solchen Benchmarklauf wird automatisch in der vorliegenden Systemumgebung mit dem kundenspezifisch festgelegten Lastprofil und vorgegebenem Transaktionsdurchsatz die maximale Anzahl performant betreibbarer Clients ermittelt. Beurteilungskriterium hierfür ist die Einhaltung einer vorgegebenen Bearbeitungszeit.

SyLaGen liefert umfangreiche Ergebnisdaten in Form von

- Konfigurationsreports aller beteiligten Systeme,
- Reports über die Ressourcen-Verbräuche aller Server-Systeme sowie
- Statistiken über die erzeugten Lastprofile und die Bearbeitungszeiten der aktiven Clients.

SyLaGen ermöglicht es, ein gegenwärtiges oder geplantes Belastungsprofil in Form einer Lastsimulation nachzubilden. So wird die Anwendung im Gesamtsystem hinsichtlich des zu erwartenden Systemverhaltens getestet. Auf der Basis des spezifizierten Belastungsprofils können unterschiedliche Hardware- und Software-Umgebungen hinsichtlich Funktionalitäts-, Verfügbarkeits- und Performanceveränderungen im Systemverhalten untersucht und verglichen werden. Darüber hinaus zeigen Stresstests das Systemverhalten während maximaler Belastung auf. Ist die konkrete Nachbildung des kundenspezifischen Belastungsprofils zu aufwändig, können vordefinierte Lastprofile für unterschiedliche Applikationen verwendet werden.

KAPITEL 06 ANFORDERUNGEN AN MESSWERKZEUGE

REINHARD BORDEWISCH, JÖRG HINTELMANN

06.01 Einführung

Ein typischer Einstieg in das Kapazitätsmanagement geschieht über die Überwachung des existierenden IT-Systems, in dem oft bereits Performance-Probleme auftreten. Hierbei wird das IT-System des Kunden zunächst vermessen und analysiert. Die Erkenntnisse münden i.d.R. in einem **Tuning** des existierenden Systems, um die Performance-Probleme zunächst kurz- bzw. mittelfristig zu lösen. Anschließend sollte eine **Planungs- und Prognosephase** folgen, um die Systemkapazitäten langfristig bedarfsgerecht zu planen. Die daraus gewonnenen Erkenntnisse sollten wenn nötig zu einer Modifikation des Systems führen, damit Performance-Probleme im laufenden Betrieb möglichst gar nicht erst auftreten.

Die **Umsetzung** dieses Vorgehensmodells erfordert Methoden und Werkzeuge, die die Aktivitäten in den einzelnen Phasen unterstützen bzw. überhaupt erst ermöglichen. Hier beschreiben wir die Eigenschaften, welche sinnvoll einsetzbare Messwerkzeuge haben sollten.

06.02 Allgemeine Anforderungen

(a) Bedienbarkeit

Zur leichteren Bedienbarkeit sind folgende Funktionen wünschenswert:

- Bedienbarkeit über Netze (remote usage, vgl. RMON-Standards)
- weitestgehend automatisierte Mess- und Auswerteumgebungen
- eindeutige Schnittstellen für nachgeordnete Weiterbearbeitung (z.B. Grafik, Modellerstellung)

(b) Experimentsteuerung

Abhängig von Zielsetzung und Zielgruppe muss Folgendes gewährleistet sein:

- Überwachung und Globalanalyse: geschieht weitestgehend automatisiert

@ Detail-/Ursachenanalyse und Ermittlung des transaktionsspezifischen Ressourcenbedarfs: ist sowohl automatisiert als auch individuell steuerbar

Problematisch ist, dass unterschiedliche Software-Monitore z.B. unter UNIX unterschiedlich gestartet werden, unsynchronisiert ablaufen und unterschiedliche, unkorrelierte Messdaten liefern. Eine umfassende Systemanalyse ist daher nahezu unmöglich.

Das Ziel besteht darin, automatisierte Performance-Messungen mit automatischer Aufzeichnung und Auswertung systemspezifischer Performance-, Last- und Konfigurationsdaten *sowie Korrelation der Messergebnisse zu ermöglichen.*

(c) Skalierbarkeit (u.a. Mess-Overhead)

Maßgeblich hierfür ist die Zielsetzung der Messung:

@ Für die Überwachung und Globalanalyse ist es wichtig, dass der Mess-Overhead und vor allem die Menge der erhobenen Messdaten nicht zu groß werden, d.h. es müssen relativ große Messintervalle und nur wenige Messindizes festgelegt werden können.

@ Dagegen verlangen Ursachenanalysen und die Ermittlung von Last- und Performance-Daten für die Prognose gezielte und detaillierte Messungen, was i.d.R. die Erhebung von vielen und exakten Messwerten über relativ kleine Zeiträume beinhaltet.

In beiden Fällen ist es erstrebenswert, die gleichen Messwerkzeuge parametrisierbar und skalierbar einzusetzen.

(d) Konsistente Globalsicht (globaler Zeitgeber)

Plattformübergreifende Sicht

Ein umfassendes Kapazitätsmanagement beinhaltet die Analyse und Bewertung des Gesamtsystemverhaltens aus Endbenutzersicht. Zur Durchführung von Messungen ist ein globaler Zeitgeber erstrebenswert, was allerdings in (weiträumig) vernetzten, heterogenen IT-Systemen schnell an Grenzen stößt. Für die weiterverarbeitenden Analyse- und Modellierungwerkzeuge müssen die Messwerkzeuge die Messdaten in einem plattformneutralen Format liefern.

Zeitliche Zuordnung

Für die Verfolgung von mehrstufigen Vorgängen ist eine eindeutige zeitliche Zuordnung der einzelnen Teilschritte notwendig (globaler systemweiter Zeitgeber). Dazu sind entsprechende Uhrensynchronisationen zu entwickeln und einzusetzen. Erst damit ist es möglich, Vorgangsdauern (Antwortzeiten) aus Teilschritten zusammenzusetzen, die auf unterschiedlichen Plattformen ablaufen.

06.03 Weitere Aspekte

(a) Unterschiedliche Messebenen (u.a. Netz: ISO-Ebenen; Gesamtsystem: Applikationen und Geschäftsprozesse)

Wünschenswert ist die Analyse des Gesamtsystems aus Endbenutzersicht. Dazu sind Messungen und Analysen des Systemverhaltens auf unterschiedlichen Systemebenen erforderlich. Im Netzwerk werden häufig Analysen auf den unteren ISO-Ebenen aber auch auf Ebene 7 durchgeführt. Bei den Server-Systemen und vor allem im Gesamtsystem sind Messungen auf unterschiedlichen physikalischen Levels und auch auf logischer Ebene von Interesse.

(b) Zuordnung von logischen Elementen zu physikalischen Ressouren

Ziel ist es, die Zuordnung von Geschäftsprozessen und den zu deren Abarbeitung erforderlichen IT-Prozessen/Transaktionen zu den physikalischen Ressourcenverbräuchen herzustellen. Dazu sind *Transaktionsverfolgung und -analyse* erforderlich: Die Messungen müssen transaktionsorientiert erfolgen. Eine Messung von Ressourcen-Verbräuchen unterhalb der Ebene von Transaktionen (Datenpakete, Prozesse, I/O-Operationen) ist nicht ausreichend.

Weiter muss eine verursachergerechte Zuordnung von Ressourcenverbräuchen möglich sein: Messungen müssen applikationsorientiert erfolgen, d.h. Ressourcenverbräuche müssen den Applikationen und Transaktionen zugeordnet werden können. Für die Kapazitätsplanung ist z.B. eine pauschale Aussage über die gesamte Bandbreitennutzung in den Netzen (LAN-Segment, Backbone, WAN) nicht ausreichend.

(c) Instrumentierbare Software

In vernetzten heterogenen Systemen gibt es nicht mehr den von der propriätären Welt bekannnten klassischen ”Transaktionsbegriff”. Damit ist auch die oben aufgeführte Zuordnung äußerst schwierig. Ein möglicher Lösungsweg ist neben der Nutzung von prozessspezifischen Messdaten (z.B. unter UNIX *ps* und *acct*) die Instrumentierung von Applikations-Software. Ansätze dazu werden u.a. mit ARM (*Application Response Time Measurement*) verfolgt, wo ein Standard API für messbare Software spezifiziert wird. Eine andere Möglichkeit wird mittels Hybrid-Monitoring angegangen.

(d) Sampling vs. Tracing (statistische Unabhängigkeit)

Sampling (Stichprobenverfahren) basiert auf Serien von Aufzeichnungen der Systemzustände zu vom Objektgeschehen unabhängigen Zeitpunkten. Die meisten klassischen Soft-

ware-Monitore liefern Übersichten darüber, wo was in welchem Zeitraum geschieht. Dabei müssen die Systembedingungen und Last relativ konstant sein, und es darf keine Abhängigkeit zwischen Stichprobe und Ereignis bestehen.

Tracing (event driven monitoring) bedeutet eine kontinuierliche Aufzeichnung von ausgewählten Ereignissen mit Angabe von Ort, Zeitpunkt, Spezifikation des Ereignisses: Tracing liefert Einsichten: Was geschieht wo, wann und warum.

Je nach Zielsetzung und Aufgabenstellung müssen die unterschiedlichen Monitore sowohl im Sampling Mode als auch im Tracing Mode arbeiten.

(e) Automatisierung der Auswertung und Aufbereitung – kriteriengesteuerte Auswertung (Filterfunktionen)

Es gibt eine Vielzahl von Mess- und Analyse-Werkzeugen, die auf unterschiedlichen Ebenen ansetzen und häufig Insellösungen darstellen. Universell einsetzbare und automatisierte Mess- und Auswerteumgebungen sind kaum vorhanden (s.o. Experimentsteuerung).

Besonders aufwändig und arbeitsintensiv sind die Auswertung der Messdaten und deren statistische Aufbereitung. Da es sich häufig um etliche GigaBytes Messdaten handelt, müssen Auswertung und die statistische Aufbereitung, aber auch die Erstellung von Präsentationstabellen und -diagrammen voll automatisiert ablaufen. Dabei muss einerseits die automatische Erstellung von Standarddiagrammen möglich sein, andererseits muss die automatische Erstellung von individuellen Diagrammen nach vorgebbaren Konfigurations-/Filterkriterien, wie Auswertezeitraum, Mittelwert- und Extremwertberechnungen oder automatische bzw. vorgebbare Korrelationsberechnungen, gegeben sein.

(f) Verlässlichkeit und Konsistenz der Messdaten

Dazu müssen alle Fehlersituationen und Ausfälle protokolliert werden, um über die Güte und Gültigkeit der Messung entscheiden zu können.

06.04 Spezielle Anforderungen hinsichtlich Überwachung

(a) Dienstgüte (Ampel-Semantik)

Für alle zu überwachenden Komponenten und Dienstgütemerkmale müssen Schwellwerte für Warnungen (gelb) und Alarme (rot) definierbar sein. Bei Erreichung der Schwellwerte sind entsprechende PopUp-Fenster zu öffnen, die den Ort und die Art der Verletzung anzeigen.

(b) Steuerung: zentral vs. verteilt

Überwachungskomponenten sollten durch eine zentrale Konsole oder durch eine Menge von Konsolen z.B. per SNMP-Nachrichten oder entsprechend RMON-Standard ein- und ausschaltbar sein. Gleiches gilt für die Abrufung von aggregierten Ergebnissen. Die Übertragung von einzelnen Ergebnissen zur Zentrale ist zu vermeiden, da sonst der Messverkehr zu groß wird.

(c) Aggregierung

Hier müssen Regel-Editoren vorhanden sein, mit denen es möglich ist, elementare Messgrößen zusammenzufassen und zu abgeleiteten bzw zu aggregierten Größen zusammenzustellen.

(d) Ergebnisdarstellung

Ergebnisse müssen sowohl in ihrer elementaren Form als auch in aggregierter Form darstellbar sein. Online-Visualisierung von Grafiken sollte ebenso möglich und wählbar sein wie die rein textuelle Darstellung der aktuellen Werte.

(e) Discovering (Suche nach aktiven Komponenten (IP-Adresse, MAC-Adressen, ...)

Überwachungswerkzeuge sollten in der Lage sein, aktive Netzkomponenten zu erkennen und in einer Liste darzustellen. Anschließend sollte es in einem Auswahlmenü möglich sein, die Überwachung einer Komponente ein/auszuschalten und die Größen zu definieren, die überwacht werden sollen. Darüber hinaus sollten Schranken definierbar sein, bei deren Überschreitung Warnungen/Alarme angezeigt werden, siehe auch Überwachung Dienstgüte (Ampelsemantik).

06.05 Spezielle Anforderungen hinsichtlich Analyse und Tuning

Neben der Sicherstellung der Betriebsbereitschaft werden Werkzeuge für ein proaktives Management benötig, die das Auftreten von Engpässen so frühzeitig erkennen, dass es zu keiner Beeinträchtigung der Leistungsfähigkeit kommt. Dabei sind die folgenden Aktivitäten durchzuführen.

(a) Ermittlung der Auslastungsgrade

Um frühzeitig Engpässe erkennen zu können, muss eine permanente Überwachung des Lastaufkommens und der Auslastung aller vorhandenen Komponenten stattfinden. Dazu ist

ein integrierter Messansatz zur automatischen Überwachung aller relevanten Systemkomponenten erforderlich.

(b) Lokalisierung von Engpässen

Bei Erkennung von Dienstgüteverletzungen muss eine Analyse hinsichtlich vorhandener Engpässe und Schwachstellen durchgeführt werden. Die Messwerkzeuge müssen in der Lage sein, aufgrund von vorgegebenen Schwellwerten die Engpässe automatisch zu erkennen und zu melden.

(c) Gezielte Ursachenanalyse

Für die gefundenen Engpässe müssen Messwerkzeuge die Ursache möglichst weitgehend automatisch ermitteln können. Mögliche Ursachen können nicht nur mangelnde Kapazitäten der genutzten IT-Konfiguration und fehlerhaftes Design der Anwendung bzw. der gesamten Architektur sondern auch organisatorische Mängel sein.

(d) Ermittlung von Wartezeiten => Transaktionsverfolgung

Es gibt auch Situationen, wo trotz niedriger Auslastung der Systemkomponenten das Systemverhalten vollkommen unbefriedigend ist. Häufig tritt derartiges in den Kommunikationsbeziehungen auf (z.B. aufgrund von großen Umschaltzeiten und langen Latenzzeiten). Die Messwerkzeuge müssen in der Lage sein, die Verweilzeiten und damit auch die Wartezeiten in den einzelnen Komponenten zu ermitteln, was letzendlich in einer Transaktionsverfolgung und -analyse mündet (s.o.).

06.06 Spezielle Anforderungen hinsichtlich Prognose

(a) Baselining (Topologie und Verkehrscharakteristik)

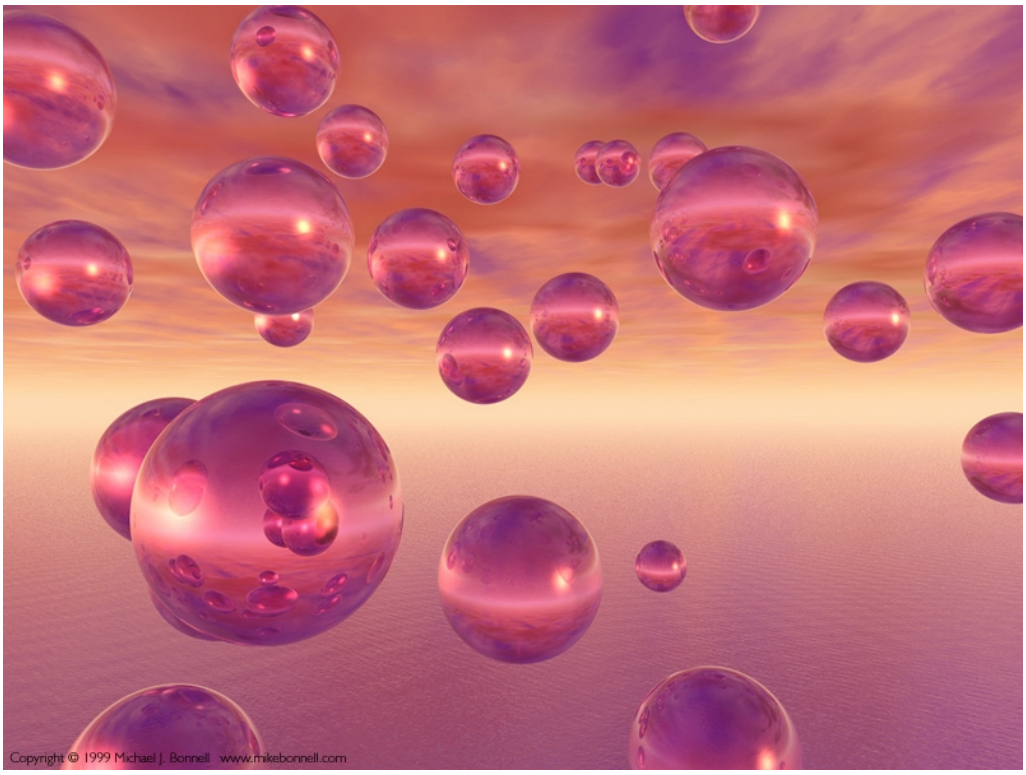
Die Messinstrumente sollten Schnittstellen besitzen, über die Informationen über Topologien und Verkehrsströme in Modellierungswerkzeuge übernommen werden können. Dabei sollten diese auf unterschiedlichen Ebenen (Linklayer, Network-Layer, Application-Layer) und je nach Modellierungswerkzeug mit unterschiedlichen Parametern übergebbar sein (Verteilungstyp, Erstes Moment, weitere Momente, ...)

(b) Übergabe aggregierter Ergebnisse an Modellierungskomponenten

Zusätzlich zu den Eingangsdaten Topologie und Verkehrsströme sollten idealerweise auch aggregierte Messergebnisse zur Validierung bzw. Kalibrierung der Modelle übernommen

werden können. Analog zu den Eingangsdaten müssen auch diese auf unterschiedlichen Anwendungsebenen (= Modellierungsebenen) übergebbar sein.

Teil III: Modellierung und Prognose



Inhalt von Teil III

KAPITEL 01 PROGNOSEMODELLE – EINE EINFÜHRUNG 4

01.01 HINTERGRUND 4
01.02 TYPISCHE FRAGESTELLUNGEN 4
01.03 VORGEHENSMODELL 5
01.04 WERKZEUGE FÜR MODELLIERUNG UND ANALYSE 5
01.05 RAHMENBEDINGUNGEN 9

KAPITEL 02 THE TOOL VITO: A SHORT SURVEY 10

02.01 SCOPE AND OBJECTIVES OF VITO 10
02.02 RESOURCE DESCRIPTION 10
02.03 WORKLOAD DESCRIPTION 10
02.04 SYNCHRONIZATION OF CHAINS 11
02.05 MODEL EVALUATION 11
02.06 CASE STUDY 11
02.07 IMPLEMENTATION OF VITO 12
02.08 USAGE OF VITO 13
02.09 REFERENCES 15

KAPITEL 03 MODELLEXPERIMENTE MIT VITO 17

03.01 EINFÜHRUNG 17
03.02 PAKETFLUSS-ORIENTIERTE MODELLIERUNG MIT COMNET PREDICTOR 17
03.03 CLIENT/SERVER-ORIENTIERTE MODELLIERUNG MIT VITO 17
03.04 ZIELE DER ARBEIT 18
03.05 AUFBAU DER ARBEIT 18
03.06 VITO: MODELLWELT UND WERKZEUG 18
03.07 MODELLIERUNG MIT VITO 21
03.08 SZENARIO: IST-SITUATION WINDOWS-TERMINAL-SERVER 27
03.09 SZENARIO: PERFORMANCE PROGNOSE BEI ERHÖHUNG DER ARBEITSLAST 28
03.10 AUSBLICK 30
03.11 LITERATUR 30

KAPITEL 04 DAS WERKZEUG ECOPREDICTOR 31

04.01 EINFÜHRUNG: ANALYTISCHE UND SIMULATIVE MODELLE 31
04.02 MODELLERSTELLUNG MIT PREDICTOR 32
04.03 SYSTEMATISCHE KAPAZITÄTSPLANUNG MIT DEM PREDICTOR 33
04.04 BEISPIEL FÜR EIN PREDICTOR-MODELL 38
04.05 ZUR ANALYSETECHNIK DES ECOPREDICTOR 40
04.06 WEITERE WERKZEUGE 41
04.07 REFERENZEN 42

KAPITEL 05 DAS SIMULATIONSTOOL COMNET-III 43

05.01 ÜBERBLICK 43
05.02 MODELLIERUNGSANSÄTZE 43
05.03 TOPOLOGIEMODELL 44
05.04 LASTMODELL 46
05.05 ERGEBNISPRÄSENTATION 48
05.06 AUSGEWÄHLTE KRITERIEN 52

KAPITEL 01 PROGNOSEMODELLE – EINE EINFÜHRUNG

BRUNO MÜLLER-CLOSTERMANN

01.01 Hintergrund

Trotz der zunehmenden Abhängigkeit des Unternehmenserfolgs von der Leistungsfähigkeit und Verlässlichkeit der IT-Strukturen ist eine systematische Kapazitätsplanung in vielen Unternehmen völlig unzureichend etabliert. Kapazitätsplanung zielt auf die Leistungsprognose für zukünftige Konfigurationen und Lastszenarien, wobei Planungen der System- und Netzinfrastruktur, z.B. die Migration von Mainframes zu C/S-Architekturen, die Einführung eines Intranets sowie die zukünftige Entwicklung der Arbeitslast in Form von neuen Anwendungen die wichtigsten Eckdaten liefern. Vereinbarungen über die gewünschte Dienstgüte wie Transaktionsdurchsatz oder Antwortzeiten bilden in Form von „Service Level Agreements“ einen weiteren wesentlichen Aspekt des Kapazitätsplanungsprozesses.

01.02 Typische Fragestellungen

Welche Auswirkungen auf Antwortzeit und Durchsatz hat ein R/3-Release-Wechsel? Wie viele gleichzeitige Nutzer kann ein Service Provider mit befriedigender Dienstgüte bedienen? Wo ist bei Einführung einer neuen Applikation der potenzielle Leistungsengpass zu erwarten? ... auf den WAN-Strecken, am File Server oder am Host? Wie ändert sich die Antwortzeit bei einer 30-prozentigen Steigerung des Transaktionsdurchsatzes?

Die Beantwortung solcher Fragen kann unmittelbar über den Erfolg eines Unternehmens entscheiden. Negativbeispiele aus derm Jahr 1999 sind der Konkurs des US-Pharmagroßhändlers Foxmeyer nach missglückter R/3-Einführung und der Rückzug von mobilcom aus dem "flat-rate"-Abenteuer, einen Internetzugang inklusive Telefongebühren für monatlich DM 77 anzubieten. Es gibt also gewichtige Gründe, Aufbau und ständige Wahrnehmung eines systematischen Kapazitätsmanagements als zentrale Aufgabe der IT-Verantwortlichen zu betrachten. Im Hinblick auf Planungsaufgaben muss das Monitoring des laufenden DV-Betriebs durch den Einsatz von Modellierungs- und Prognosewerkzeugen ergänzt werden, so dass Konfigurierung und Sizing von Server- und Netzwerkkomponenten unter zukünftigen Szenarien durch Modellprognosen begleitetet und unterstützt werden können. Geeignete Prognosetools haben sich in den USA etabliert und werden nun zunehmend auch im europäischen Raum angeboten.

01.03 Vorgehensmodell

Eine systematische Kapazitätsplanung unterscheidet die folgenden vier Phasen.

- **Festlegung von Rahmenbedingungen:** Diese Phase dient dazu, im Unternehmen die entsprechenden Vorbereitungen zu treffen, u.a. bedeutet dies die Akzeptanz beim beteiligten Personenkreis zu fördern, Unternehmensziele und Geschäftsprozesse zu erfassen und schließlich Aktionsplan und Kostenrahmen festzulegen.
- **Erfassung und Darstellung der Ist-Situation:** Dazu gehört die Bestandsaufnahme von IT-Komponenten (Ressourcen aller Art), die Erfassung der IT-Prozesse sowie die Abbildung von Geschäftsprozessen auf die IT-Prozesse. Das aktuelle Leistungsverhalten im System- und Netzbereich sowie die Last- und Verkehrscharakteristik bilden die Grundlage zur Entwicklung eines sog. Basismodells. Die Validation des Basismodells durch Abgleich mit Mess- und Modelldaten erhöht das Vertrauen in die späteren Prognosen.
- **Planung zukünftiger Szenarien:** Ausgehend von den Unternehmenszielen wird die künftige Entwicklung der unternehmenskritischen Geschäftsprozesse definiert. Dies führt zu Aussagen über das erwartete Lastverhalten, was eine wesentliche Vorraussetzung zur Entwicklung von Prognosemodellen bildet.
- **Einsatz von Prognosemodellen und Bewertung der Soll-Situation:** Als Ergebnis der Systemplanung enstehen Konfigurationsalternativen, die unter verschiedenen Lastszenarien auf ihre Tauglichkeit untersucht werden. Wesentliches Bewertungskriterium sind dabei Dienstgütevereinbarungen, wie z.B. eine mittlere Antwortzeit von maximal 2 sec, oder ein Transaktionsdurchsatz von 15000 TA/h. Spezielle Analysewerkzeuge dienen der Erstellung und Lösung der Modelle.

01.04 Werkzeuge für Modellierung und Analyse

Der Begriff Leistungsprognose impliziert den Einsatz von mathematisch-analytischen oder auch simulativen Modellen, mit welchen Vorhersagen zum Leistungsverhalten von IT-Systemen gemacht werden können. Der Planungsprozess muss daher auch quantitative Informationen einschließen.

Tools zur Kapazitätsplanung bieten Möglichkeiten zum Import externer Daten in die Analysewerkzeuge, so können z.B. aus Messdaten von System- und Netzwerkmanagement-Werkzeugen automatisch Basismodelle erstellt und anschließend für unterschiedliche Lastszenarien ausgewertet werden. Beispiele für kommerziell verfügbare Werkzeuge sind der COMNET Predictor für die Netzwerkplanung, BEST/1 von BMC Software, der Strategizer von SES für Client/Server-Kapazitätsplanung sowie die OPNET-Tools von MIL3.

Werkzeug	Distributor	Einsatzbereich	Methode
BEST/1	BMC (einst BGS)	Mainframes	Math. Analyse
Predictor	Compuware (einst CACI)	Paket-Netze	Math. Analyse
COMNET-III	Compuware (einst CACI)	vielfältig, durch add-ons	Simulation
SES/strategizer	HiPerformix (einst SES)	Client/Server	Simulation
OPNET Planner	MIL3	Netze	Simulation
OPNET Modeller	MIL3	vielfältig, durch add-ons	Simulation
VITO	Uni Essen & Siemens	Allgemein	Math. Analyse
WLPSizer	Uni Essen & Siemens	R/3-Systeme	Math. Analyse

Tabelle 01-1: Werkzeuge zur Kapazitätsplanung

Um den Einstieg in den Bereich der Modellprognose zu erleichtern, bieten die Werkzeug-Distributoren z.T. kostenfreie Einführungskurse und Evaluierungslizenzen. Die Einstiegspreise für eine Industrielizenz reichen von ca. 20000 Euro (Predictor, OPNET Planner) bis ca. 40000 Euro (COMNET-III Simulator, SES/strategizer, OPNET Modeler/Radio). Bei einer Kaufentscheidung ist zu berücksichtigen, ob die vorhandenen Mess- und Monitoringmöglichkeiten die Modellentwicklung ausreichend unterstützen. Insbesondere ist auch zu bedenken, dass die Parametrisierung und Auswertung von Simulationsmodellen im allgmeinen sehr aufwändig ist. Die analytischen Werkzeuge wie BEST/1 und COMNET Predictor erlauben aufgrund der stärkeren Abstraktion das schnelle interaktive Durchspielen von zahlreichen Modellvarianten und sind daher bedeutend effektiver als die simulativen Werkzeuge.

(a) BEST/1

BEST/1 ist ein Werkzeug, das sich im Mainframe-Bereich seit über 20 Jahren erfolgreich behauptet und seitdem kontinuierlich weiter entwickelt hat. Der Urheber von BEST/1, Firmengründer und Chef-Theoretiker Jeff Buzen, verdankt seinen Geschäftserfolg der Fokussierung auf Großkunden mit geschäftskritischen interaktiven Diensten; in der Historie waren dies vor allem Fluggesellschaften und Banken. Außerdem wurden schon früh problemangepasste Bedienoberflächen und die enge Verzahnung von SNMP-basierten Mess- und Monitoringmethoden mit Modellierungstechniken realisiert. Messdaten werden an die Auswerte- und Visualisierungskomponenten BEST/1-Monitor und Visualizer übermittelt und zur Konstruktion und Auswertung von Prognosemodellen verwendet. Die Auswirkungen von Konfigurations- und Laständerungen können dann per Modellanalyse durch die Komponente BEST/1-Predict berechnet werden. Neben der klassischen Mainframe-

orientierten Variante des Werkzeugs existiert inzwischen auch BEST/1 for Distributed Systems. Die Übernahme von BEST/1 durch BMC Anfang 1998 zeigt den Trend des Zusammenwachsens von Modellierungs- und Prognosewerkzeugen.

(b) Predictor und COMNET-III-Simulator

Der Predictor ehemals von CACI, heute vertrieben durch Compuware als Teil der EcoTool Suite, ist ein Werkzeug, welches auf die Analyse von Netzwerken spezialisiert ist. Predictor-Modelle sind aufgebaut wie die Pläne der System- und Netzadministration, d.h. es gibt Icons für Router, Switches, LANs, Subnetze, Hosts, Server und Clientgruppen. Die Verkehrslast wird durch sog. Paketflüsse spezifiziert, welche sich gemäß der gängigen Routingstrategien (z.B. kürzeste Wege) durch das Netz bewegen. Die besondere Herausforderung für den Anwender besteht darin, sinnvolle Parametrisierungen von Modellbausteinen und Lastbeschreibungen zu ermitteln. Mit dem Partnerwerkzeug COMNET Baseliner können u.a. automatisch erfasste Topologiedaten als Grundlage für Modellentwicklungen genutzt werden. Die Stärken des Predictors sind seine Schnelligkeit – mehrere Vorhersageperioden werden in Sekundenschnelle berechnet – und die detaillierte und zielgerichtete Auswertemöglichkeit. Es können z.B. Auslastungen und Durchsätze für alle Komponenten getrennt nach Szenarien ermittelt werden. Differenziertere, aber wesentlich aufwändigere Modellexperimente können mit dem Simulator COMNET-III durchgeführt werden. Der Simulator ermöglicht eine wesentlich detailliertere Beschreibung von System- und Lasteigenschaften, wobei der höhere Aufwand für die Erstellung, Auswertung und Pflege der Modelle berücksichtigt werden muss. Für den COMNET-III-Simulator werden eine Reihe von spezialisierten Add-On-Packages, so z.B. für Client/Server-Systeme, für ATM-Netze und für Satelliten-Kommunikation angeboten.

(c) SES/strategizer

Dem COMNET-III-Simulator vergleichbar ist der SES/strategizer, welcher von vornherein als Client/Server-Werkzeug konzipiert ist. Die Parameter aller Bausteine, wie z.B. Angaben zur Leistungsfähigkeit von Prozessoren oder Netzkomponenten, sind Bestandteil der Distribution und müssen nicht als separates Modul hinzugekauft werden. In jedem Falle sind diese Bausteine aber permanent zu pflegen, zu validieren und bez. des Einsatzbereiches anzupassen. Nur dann sind vertrauenswürdige Prognoseergebnisse zu erzielen. Die Problematik der Modellvalidation ist eine dem modellgestützten Ansatz inhärente Eigenschaft, mit der alle Werkzeuge gleichermaßen konfrontiert sind.

(d) Die OPNET-Tools

Die Werkzeuge OPNET Planner und OPNET Modeler von MIL3 bieten ähnliche Funktionalitäten wie die bereits genannten Tools. Der Planner ist vorwiegend für die schnelle Si-

mulation von Netzwerken aus vorgefertigten Bausteinen konzipiert, während der Modeler zusätzlich die Definition von modifizierten und benutzerspezifischen Bausteinen zulässt. Insbesondere ist auch die hierarchische Spezifikation von großen, komplexen Systemen bis hin zur Beschreibung von Protokollsoftware explizit möglich. In umfangreichen Bibliotheken werden u.a. Modellbausteine angeboten für Protokolle, Netzwerkkomponenten und Verkehrsquellen. Beispielweise gibt es in Form des OPNET Modeler/Radio eine Version, welche Bausteine speziell für die Analyse von Mobilnetzen mit Radio- und Satellitenverbindungen enthält. Durch eine strategische Allianz von MIL3 und HP versprechen die OPNET Tools besonders gut mit HP OpenView und HP NetMetrix zu kooperieren.

(e) VITO

VITO beruht auf geschlossenen Warteschlangennetzen, welche auch für zahlreiche Stationen und umfangreiche Anzahlen von Lastketten mit effizienten Algorithmen gelöst werden können[1]. Die Zeit für eine Analyse[2] liegt im Sekundenbereich, wobei für Modelle größerer Komplexität die Rechenzeit im Bereich von einer bis mehreren Sekunden liegen kann.

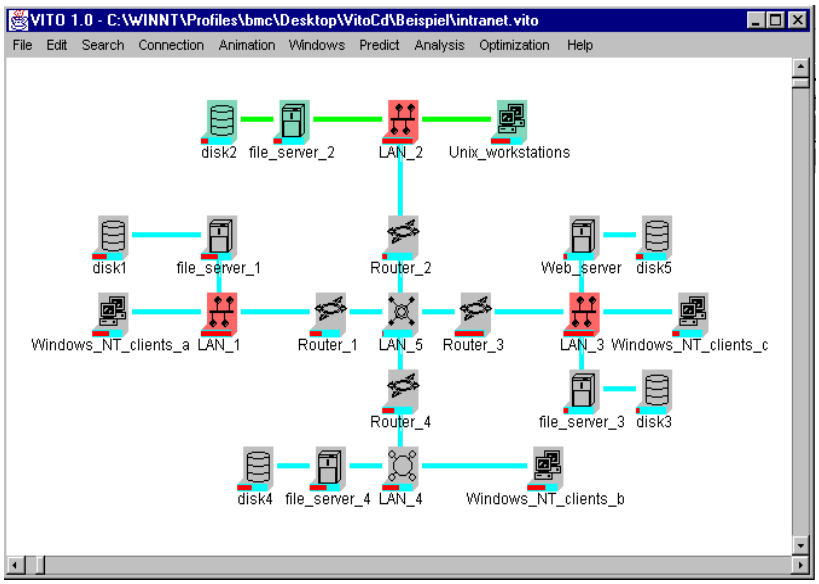


Abbildung 01-1: Layout eines typischen VITO-Modells

[1] In VITO wird der approximative Bard/Schweitzer-Algorithmus verwendet.
[2] Es handelt sich um die Berechnung von Mittelwerten für Leistungsmaße, wobei stationäres Systemverhalten vorausgesetzt wird.

VITO bietet Bausteine an, welche bereits mit typischen Werten parametrisiert sind, z.B. Fast Ethernet und Token-Ring oder Multiprozessor-Systeme und Plattensysteme. Diese Bausteine bilden die Ressourcen des Systems, welche von der Arbeitslast in Form von sogenannten Lastketten in Anspruch genommen werden. Die Abbildung 01-1 zeigt die grafische Darstellung eines VITO-Modells für ein Intranet [Mena98].

Das in dieser Abbildung dargestellte Intranet besteht aus Servern, Plattensystemen, Workstations und Netzkomponenten. Die Parameter der Stationen beschreiben die Leistungsfähigkeit des Systems; die Lastbeschreibung spezifiziert die Ressourcenanforderungen eines bestimmten Anwendungsszenarios, hier z.B. eine Multi-Media-Trainingsumgebung.

(f) WLPSizer

Der *WLPSizer* („Workload Profile Sizer“) ist ein Werkzeug zur Kapazitätsplanung von SAP R/3 Client/Server-Systemen. Als Eingabe erhält der *WLPSizer* Beschreibungen von R/3-Konfigurationen und Arbeitslasten. Die zur Lastbeschreibung benötigten Informationen liefert das Softwareprodukt „myAMC.LNI“. Zur Lastbeschreibung verwendet der *WLPSizer* Workloads (Lastprofile), wobei die Dialogschritte klassifiziert werden nach Typ (Dialog, Update, Batch, Others) und ihrer Komplexität (1 = sehr einfach, ..., 5 = sehr komplex). Der *WLPSizer* berechnet aus den gegebenen System- und Lastdaten die jeweiligen Durchsätze, Auslastungen, Antwort- und Verweilzeiten, indem die R/3-Konfiguration in ein Warteschlangennetz abgebildet und mit Hilfe sehr effizienter analytischer Algorithmen gelöst wird. Mit Hilfe des *WLPSizer* können somit SAP R/3 Konfigurationen unter beliebigen Lastszenarien bewertet werden.

01.05 Rahmenbedingungen

Damit in einem Unternehmen die für das Kapazitätsmanagement erforderlichen Prozesse bis hin zu vertrauenswürdigen Prognosemodellen und gesicherten Planungsentscheidungen erfolgreich ablaufen können, sind seitens des IT-Managements die notwendigen Rahmenbedingungen zu schaffen, wobei das zu betrachtende Spektrum von den Unternehmenszielen, über Geschäftsprozesse, Anwendungsentwicklung, System- und Netzadministration bis hin zum Einsatz von Modellierungs- und Prognosewerkzeugen reichen sollte. Zum erfolgreichen Einsatz von Prognosewerkzeugen für die Kapazitätsplanung ist die Zusammenführung unterschiedlicher Fachkompetenzen in einem kooperativ arbeitenden Team in jedem Falle von wesentlicher Bedeutung.

KAPITEL 02 THE TOOL VITO: A SHORT SURVEY

MICHAEL VILENTS, CORINNA FLÜS, BRUNO MÜLLER-CLOSTERMANN

02.01 Scope and Objectives of VITO

The queueing network tool VITO has been developed to support system and network designers during important stages of the capacity planning process. In particular, VITO serves to describe models in a style that is close to the system and network diagrams used by adminstrators and planners. Resources like servers, clients, LANs, WANs, and network devices are available as building blocks. Server capacity and usage of resources is described in real-life units, e.g. MBit/sec, packets/sec, TA/h or user think time. If the model predicts violations of service levels like response time $\geq$ 3 sec or utilization $\geq$ 85 % this is highlighted in the main window. Due to the fast approximate queueing network algorithms series of model experiments over a wide range of parameters may be performed interactively within a very short time.

02.02 Resource Description

The VITO interface offers a number of building blocks that are mapped to the stations of a closed queueing network. The station types include PS, IS, QD and FCFS disciplines. A station with QD-discipline (QD = queue dependent) serves its customers with a speed factor that depends on the number of present customers; important applications of QD-stations are the modellings of multi servers or networks. Each building block has a basic speed that is expressed in terms like jobs/sec (host), Mbit/sec (LAN) or Mbyte/sec (file server). The resources may be connected to show the system's physical topology.

02.03 Workload Description

The workload is described by tasks (customers, jobs, packets, ...) belonging to closed chains as known from queueing network theory. The service demand is described in units that are compatible with the speed units of the building blocks, eg. a service demand of 125 Kbyte at a 10 Mbit-network will yield a time demand of 0.1 sec.

Instead of a routing of tasks through the model the notation of visit counts is used to quantify the relative visit frequency directly.

02.04 Synchronization of chains

A special and very useful feature is the synchronization of workload chains by artificially introducing delays for "fast chains" that have to wait for the slower ones. As described by Schätter and Totzauer this feature can be integrated in queueing network algorithms [5], [6]. Indeed VITO uses the algorithms implemented by G. Totzauer as described in [5]. From the user viewpoint chains can be related (non-recursively) to each other, e.g. such that chain i has the multiple throughput of chain j. A typical application is the synchronization of chains with respect to their throughput, e.g.: "The throughput of print-jobs must be equal to the throughput of a certain type of tasks". Another example is the modelling of a complex applications that consists of tasks A, B and C. Such an application is completed if A, B and C have completed one, two and three cycles, respectively. This can be accomplished by enforcing a throughput relation of 1:2:3.

02.05 Model Evaluation

Model results are shown as tables, in which the user can switch between different viewpoints. Graphical visualization and statistical investigation of data can be done by exporting .csv or .txt files. Another feature of VITO is the interactive evaluation of experiment series that allows the fast investigation of parameter dependencies.

02.06 Case Study

A large provider serves about 100 remote locations with 5000 users. The main application (APP) is to be migrated from a classical host system to a client/server architecture. The encapsulation of the host application will provide the same functions as before. Additionally on the client PCs the new graphical interface (GUI) requires additional 50 Kbyte per APP transaction (in the average) to be moved from a file server to the clients. Moreover, users have access to an online-manual (MAN) that is provided by an intranet server. The access rate to the online-manual is assumed to be proportional to the transaction throughput. In the average, for 20 transactions there is one access to the manual. The file server and the intranet server both are able to put 2 MByte/s on the network, i.e. they have a speed of 2 MByte/s. The host has a mean delay of 0.06 s per APP transaction. The clients are connected with the servers via a 10BaseT ethernet and an FDDI ring.

The three transaction types APP, GUI and MAN are modelled by six separate chains: One chain for each transaction type which is produced by a client group of 30 active users (client group), and aggregated chains for each transaction type produced by the rest of clients (aggregated client group). The performance analysis is focused on the client group.

The throughput ratios of the chains concerning the client group are given by 1:1:0.05. The same applies to the chains for the aggregated client group. VITO allows to specify these throughput ratios as model input (see Figure 02-3). As a consequence the "fast" chains will be delayed to meet the specified throughput relations. In this case the workload each chain produces is given by the transferred amount of data.

The physical system structure is modelled by VITO‘s stations and the frequency a chain visits a station. Although the workload‘s, i.e. the chain’s routes through the system are irrelevant for the underlying algorithms, VITO allows to draw connections between the stations for a higher clearness of the system topology. Thus the system topology can be built as shown in Figure 02-1.

After building the model, VITO’s execute function computes the stationary solution and a result window opens up. In the case shown in Figure 02-4 it is evident that the file server has a capacity problem: its utilization is nearly 100 %. As a consequence the response time of the chain which passes through the file server (GUI) is intolerable high. Hence, the file server‘s speed is too slow for the actual workload. The analysis function can now be used to examine the file server’s utilization or the response time of GUI depending on the file server’s speed.

02.07 Implementation of VITO

VITO‘s graphical user interface has been implemented in JAVA using Visual Cafe. The state described here has been reached with an implementation effort of about eight person months. The algorithms (about one thousand lines of C Code) implement a modified Bard/Schweitzer approach including queue-dependent stations [5]. Since early 1999 VITO has been developed with the partial support of SIEMENS AG in the MAPKIT-Project funded by the *bmb+f*[3].

[3] Bundesministerium für Bildung und Forschung

02.08 Usage of Vito

(a) Main Window

The main window gives an overview of the modelled system. In the upper left part chains and appropriate stations are listed. In the lower left part the chains are grouped by the stations. The right part shows the system topology. Users can arrange the stations as they like and connect them via lines to improve the visualization.

Marking one chain and double-clicking one station the chain specific station parameters (service amount, visit count) can be edited.

The main window also denotes the station's utilization (red bar at station File_Server in Figure 02-1). A longer bar indicates a higher utilization.

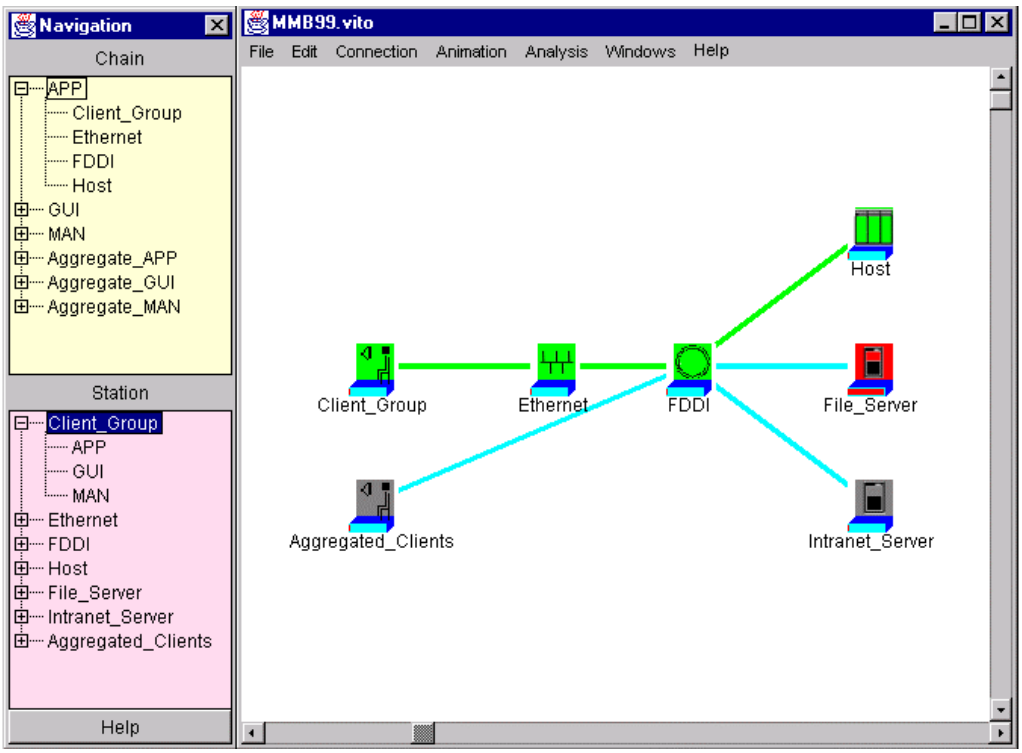


Figure 02-1: Main Window

(b) Edit Stations and Chains

The Edit Station menu defines the chain independend station parameters: station type, speed (several units can be choosen), and the icon which represents it in the main window.

The Edit Chain menu allows the definition of station independend chain parameters: population and relation among several chains e.g. according to throughput ratios. Graphical details are shown in Figures 02-2 and 02-3.

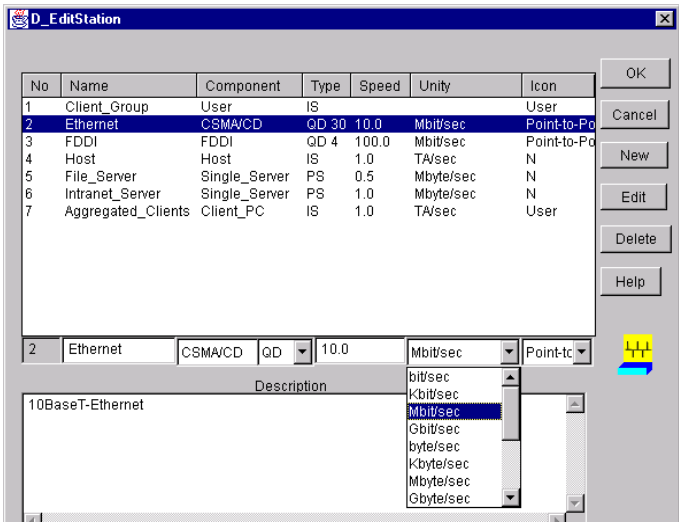


Figure 02-2: Edit-Station Menu

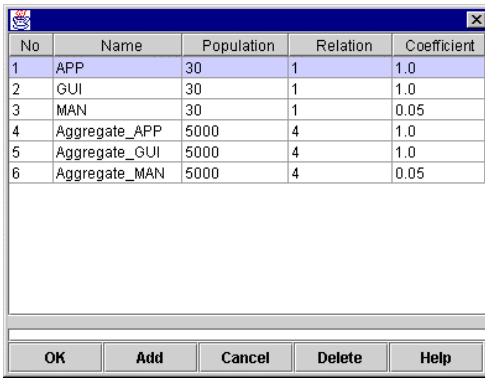
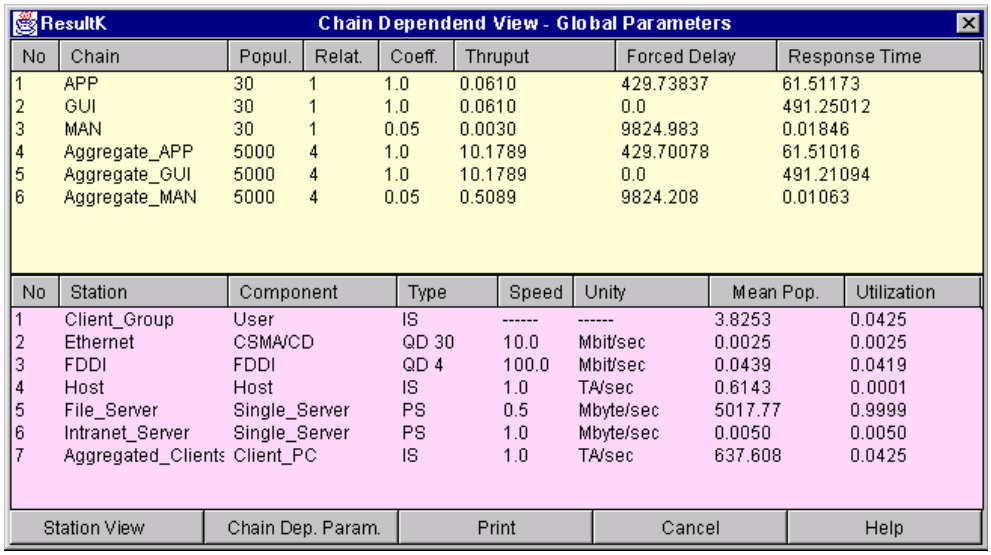


Figure 02-3: Edit-Chain Menu

(c) Results

After model's stationary solution results are shown as in Figure 02-4. The user can choose between a chain oriented, a station oriented or a global view.

The „forced delay" refers to the retarding of faster chains which are synchronized with slower ones.



No	Chain	Popul.	Relat.	Coeff.	Thruput	Forced Delay	Response Time
1	APP	30	1	1.0	0.0610	429.73837	61.51173
2	GUI	30	1	1.0	0.0610	0.0	491.25012
3	MAN	30	1	0.05	0.0030	9824.983	0.01846
4	Aggregate_APP	5000	4	1.0	10.1789	429.70078	61.51016
5	Aggregate_GUI	5000	4	1.0	10.1789	0.0	491.21094
6	Aggregate_MAN	5000	4	0.05	0.5089	9824.208	0.01063

No	Station	Component	Type	Speed	Unity	Mean Pop.	Utilization
1	Client_Group	User	IS	------	------	3.8253	0.0425
2	Ethernet	CSMA/CD	QD 30	10.0	Mbit/sec	0.0025	0.0025
3	FDDI	FDDI	QD 4	100.0	Mbit/sec	0.0439	0.0419
4	Host	Host	IS	1.0	TA/sec	0.6143	0.0001
5	File_Server	Single_Server	PS	0.5	Mbyte/sec	5017.77	0.9999
6	Intranet_Server	Single_Server	PS	1.0	Mbyte/sec	0.0050	0.0050
7	Aggregated_Clients	Client_PC	IS	1.0	TA/sec	637.608	0.0425

Figure 02-4: The Result Window

02.09 References

[1] H. Beilner (ed.); Werkzeuge zur modellgestützten Leistungsbewertung , it+ti (Informationstechnik und Technische Informatik), Schwerpunkthema, Heft 3/95, Oldenbourg-Verlag

[2] Bolch et al.; Leistungsbewertung mit PEPSY-QNS und MOSES, in [1]

[3] MAPKIT-Server: Projektserver zum Thema Kapazitätsmanagement, http://mapkit.uni-essen.de

[4] D. Menasce, V.A.F. Almeida, L.W. Dowdy; Capacity Planning and Performance Modelling: From Mainframes to Client Server Systems, Prentice Hall, 1994

[5] A. Schätter; G. Totzauer; Aufteilung von Rechenkapazität durch Steuerung von Durchsätzen, in H. Beilner (Hrsg.): Proceedings ‚Messung, Modellierung und Bewertung von Rechensystemen', Springer-Verlag, 1985

[6] G. Totzauer; Kopplung der Kettendurchsätze in geschlossenen Warteschlangennetzwerken, in U. Herzog and M. Paterok (Hrsg.): Proceedings ‚Messung, Modellierung und Bewertung von Rechensystemen', Springer-Verlag, 1987

KAPITEL 03 MODELLEXPERIMENTE MIT VITO

BRUNO MÜLLER-CLOSTERMANN, MICHAEL VILENTS

03.01 Einführung

Eine Unternehmung aus dem öffentlichen Bereich betreibt ein komplexes, heterogenes Client/Server-System, welches für einige Tausend Benutzer sehr unterschiedliche Dienste zur Verfügung stellt. Eine ausführliche Darstellung der Situation inklusive des Planungsstandes war vorhanden, inklusive Informationen zur Netztopologie, zum Netzverkehr, zum Typus der meistgenutzten Applikationen und zur Funktionsweise der Client/Server-Interaktionen. Hierbei sind insbesondere das ICA-Protokoll für die Windows-Terminal-Server (WTS) und die ICA-Clients besonders zu beachten.

03.02 Paketfluss-orientierte Modellierung mit COMNET Predictor

Die durchgeführten Messungen quantifizieren Intensität und Volumen des Paketverkehrs. Unter Verwendung dieser Informationen wurde ein Netzmodell erstellt [Alan2000], in welchem sich Paketflüsse von Client-Rechnern (Origins) zu Server-Systemen (Destinations) und zurück bewegen. Diese Art der Modellbildung war Grundlage für eine analytische Lösung mit dem Modellierungs- und Analyse-Tool Comnet Predictor[4], mit welchem Auslastungen, Durchsätze und Paketverzögerungen berechnet werden können. Das verwendete Modellierungsparadigma beruht auf paketvermittelnden Feedforward-Netzen (ohne Verzweigungen, ohne Rückkoppelungen, ohne Zyklen), durch welche die Paketflüsse von einer Quelle zu einem Ziel geroutet werden.

03.03 Client/Server-orientierte Modellierung mit VITO

In dieser Arbeit wird ein anderer Ansatz verfolgt, bei dem die Arbeitslast nicht nur auf Paket- und Netzebene, sondern auch durch die Ressourcen-Anforderungen an Servern und I/O-Systemen beschrieben wird. Die Arbeitslast besteht also aus Lasteinheiten, welche zyklisch die unterschiedlichen Ressourcen belasten und zusätzlich insbesondere das Benutzerverhalten mit berücksichtigen. Dadurch können auch Rückkoppelungseffekte zwischen Benutzern und dem System erfasst werden. Dieser Ansatz ist in dem Modellierungs- und

[4] Seit Ende 2000 wird das Werkzeug Comnet Predictor unter dem Namen EcoPredictor geführt.

Kapazitätsplanungswerkzeug VITO implementiert, welches an der Universität Essen in den Jahren 1999 und 2000 entstanden ist [VITO2000][5].

03.04 Ziele der Arbeit

Diese Arbeit demonstriert, dass der durch das Werkzeug VITO unterstützte Ansatz geeignet ist, Performance-Analysen und Prognosen für komplexe, reale Installationen durchzuführen, so dass Netz- und Systemplaner bei einem systematischen Kapazitätsmanagement sinnvolle Entscheidungshilfen zur Hand haben. Insbesondere die Einhaltung von Dienstgütevereinbarungen, wie z.B. Dialog-Antwortzeit im Mittel < 2 sec oder systembezogene Größen wie Server-Auslastung < 80 %, stellen ein wichtiges Kriterium für die Bewertung von Konfigurations-Alternativen dar. Der Fokus liegt auf der Modellierung von typischen Strukturen von Client/Server-Systemen, wobei eine konkrete Installation sowie Messergebnissen zusammen mit plausiblen Annahmen zugrunde gelegt werden. Neben der Vorstellung der allgemeinen Methodik wird konkret in die toolgestützte Modellbildung und Analyse mit VITO eingeführt.

03.05 Aufbau der Arbeit

Die Arbeit ist folgendermaßen aufgebaut. Zunächst werden in Abschnitt 03.06 die VITO-Modellwelt und das VITO-Werkzeug anhand eines Beispiels übersichtsartig vorgestellt. Abschnitt 03.07 beschreibt wie System- und Lastbeschreibung für das DVV-System organisiert werden können. Die folgenden Abschnitte 03.08 und 03.09 skizzieren einige typische Szenarien und die zugehörigen Analyse-Ergebnisse.

03.06 VITO: Modellwelt und Werkzeug

VITO beruht auf geschlossenen Warteschlangennetzen, welche auch für zahlreiche Stationen und umfangreiche Anzahlen von Lastketten mit effizienten Algorithmen gelöst werden können[6]. Die Zeit für eine Analyse[7] liegt im Sekundenbereich, wobei für Modelle größerer Komplexität die Rechenzeit im Bereich von einer bis mehreren Sekunden liegen kann. VITO bietet Bausteine an, welche bereits mit typischen Werten parametrisiert sind, z.B. Fast Ethernet und Token-Ring oder Multiprozessor-Systeme und Plattensysteme. Diese

[5] VITO entstand im Rahmen des vom BMBF geförderten Verbundprojektes MAPKIT, an welchem die Arbeitsgruppe an der Universität Essen sowie die Unternehmen Fujitsu Siemens Computers, Siemens Business Services und Materna Information and Communications beteiligt waren.

[6] In VITO wird der approximative Bard/Schweitzer-Algorithmus verwendet, vgl. [Jain1991, Mena98].

[7] Es handelt sich um die Berechnung von Mittelwerten für Leistungsmaße, wobei stationäres Systemverhalten vorausgesetzt wird.

Bausteine bilden die Ressourcen des Systems, welche von der Arbeitslast in Form von sogenannten Lastketten in Anspruch genommen werden. Die folgende Abbildung 03–1 zeigt die grafische Darstellung eines VITO-Modells für ein Intranet [Mena98].

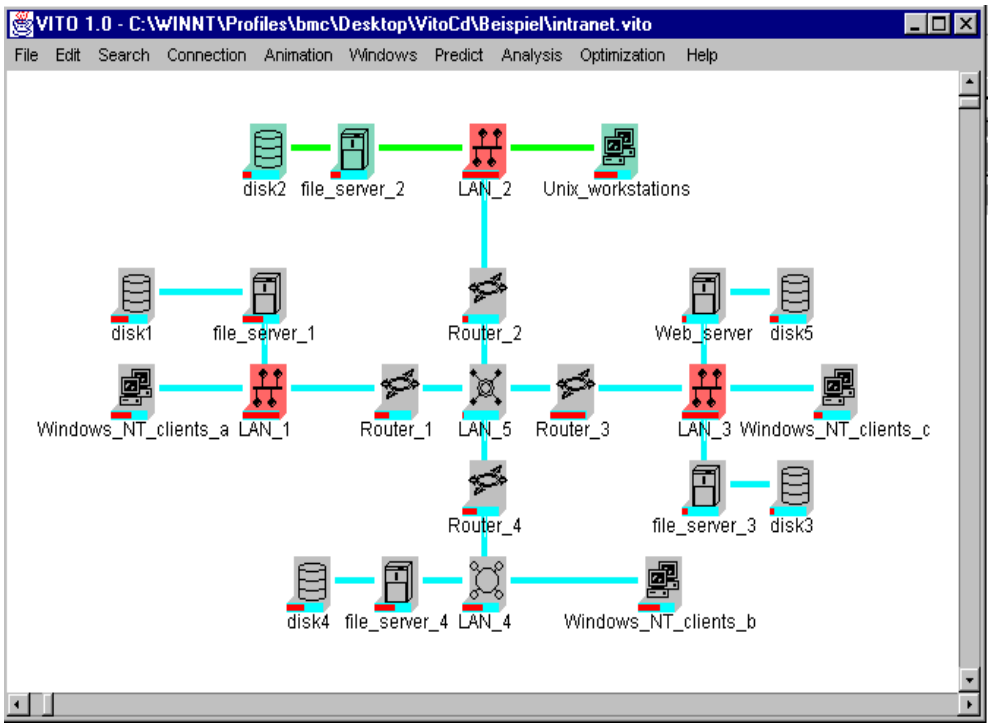


Abbildung 03-1: Layout eines typischen VITO-Modells (frei nach Menascé 1998)

Das in Abbildung 03-1dargestellte Intranet besteht aus Servern, Plattensystemen, Workstations und Netzkomponenten. Die Parameter der Stationen beschreiben die Leistungsfähigkeit des Systems; die Lastbeschreibung spezifiziert die Ressourcenanforderungen eines bestimmten Anwendungsszenarios, hier z.B. eine Multi-Media-Trainingsumgebung. Details finden sich in dem Lehrbuch zur Kapazitätsplanung von D. Menasce [Mena2000].

Nach erfolgter Auswertung werden überlastete Stationen[8] optisch hervorgehoben. Hier ist dies für die lokalen Netze LAN1, LAN2 und LAN3 der Fall. Nach Austausch der Ethernet Bausteine (10 Mbit/sec) durch Fast Ethernet Bausteine (100 Mbit/sec) verschiebt sich der

[8] Für Auslastung, Antwortzeit und Durchsatz können Service Levels definiert werden, deren Einhaltung bzw. Verletzung geprüft wird.

Engpass von den LAN-Segmenten weg, hin zu anderen Stationen. Eine detaillierte Untersuchung ist z.B. mit der Predict-Funktion von VITO möglich, welche hier pro Periode eine 10%-ige Laststeigerung (bezogen auf die Client-Anzahl) über einen Zeitraum von 24 Perioden analysiert.

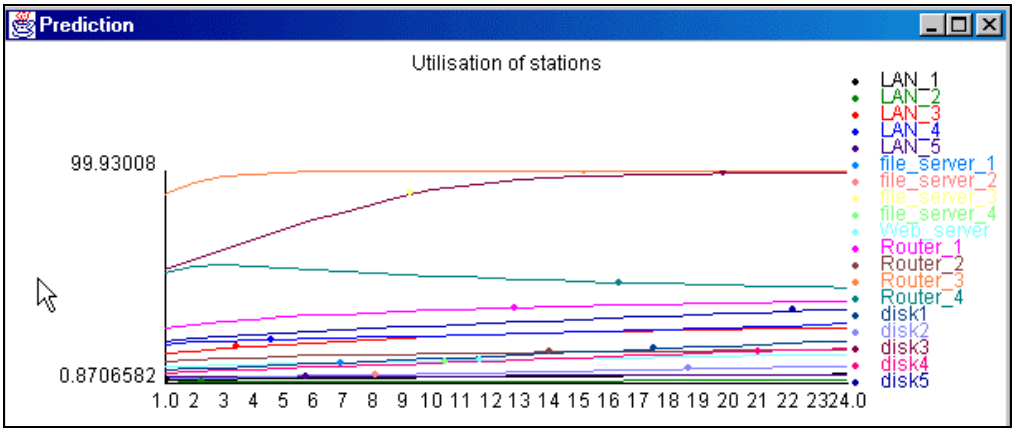


Abbildung 03-2 : Auslastung über 24 Perioden bei Laststeigerungen von 10 % pro Periode

Die Abbildung 03-2 zeigt, dass mehrere Stationen, vor allem Router_3 und sowie disk3 und auch Router_4 (von oben nach unten), bereits unter geringem Lastzuwachs Engpässe darstellen. Vor weiteren Untersuchungen sind Vorschläge zur Beseitigung dieser Engpässe zu erarbeiten. Nach Erhöhung der Routerkapazität von 1000 auf 5000 Pakete pro Sekunde und Verdoppelung der DISK-I/O-Kapazitäten ergibt sich folgendes Bild, vgl. Abbildung 03-3.

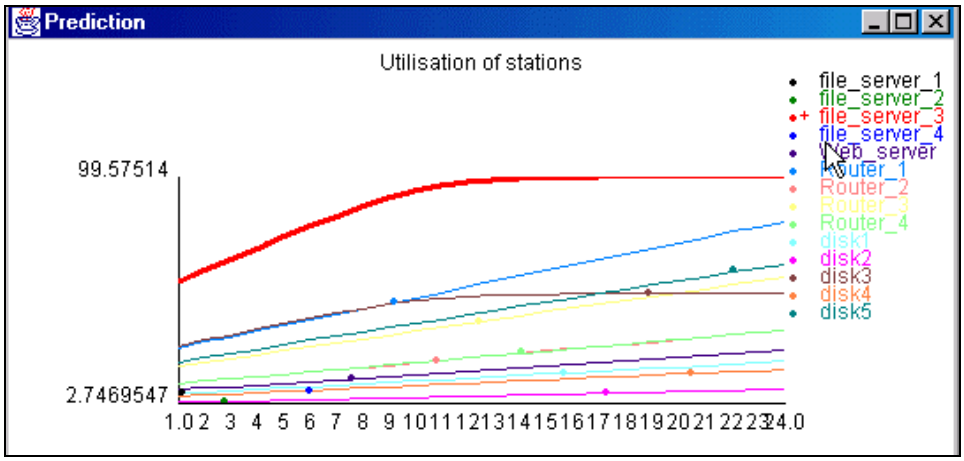


Abbildung 03-3: Die Station file_server_3 wird zum Engpass

Weitere Modellrechnungen können z.B. das Antwortzeitverhalten berücksichtigen oder alternative Lastszenarien durchspielen.

03.07 Modellierung mit VITO

Es ist hier nicht das Ziel, die paketorientierte Modellierung wie sie z.B. im COMNET Predictor verwendet wird mit VITO zu reproduzieren, sondern es soll gezeigt werden, wie eine Modellierung auf Applikationsebene unter Berücksichtigung der Netzstruktur und des Paketverkehrs angegangen werden kann. Dabei wird unter anderem deutlich werden, welche Messgrößen zu diesem Zweck erhoben werden müssen. Die mit VITO durchgeführte Modellierung betrachtet verschiedene Szenarien, wobei bezüglich der Parameter auf Applikations- und Server-Ebene plausible Annahmen gemacht wurden. Die Information über die Paketflüsse auf Netzebene, wie z.B. durchschnittliche Paketgrößen wurde in Anlehnung an die durchgeführten Messungen gewählt.

(a) IST-Analyse

Die Modellbildung setzt eine sog. **Ist-Analyse** voraus[9], welche neben einer Bestandsaufnahme der gesamten Netzwerk- und Server-Infrastrukur sowie der Hauptapplikationen auch quantitative Aussagen über die Arbeitslast, die Ressourcenauslastung und die aktuelle Dienstgüte umfasst.

[9] Eine IST-Analyse mit Schwerpunkt auf der Netzstruktur und dem Paketverkehr wurde in der Studie [GLHAG2000] durchgeführt.

(b) Basismodellierung

Auf Grundlage der Ist-Analyse wird zunächst ein sogenanntes **Basismodell** entwickelt, welches die aktuelle Situation widerspiegelt. Dies bedeutet, dass alle wesentlichen Hardware- und Softwarekomponenenten berücksichtigt sind und die mit Hilfe des Modells berechneten Performancegrößen mit den gemessenen Werten übereinstimmen. Meist ist es sinnvoll, sich auf die wesentlichen Komponenten und die Hauptapplikationen zu konzentrieren und nur diese in angemessener Detailliertheit ins Modell aufzunehmen. Der „Rest" des Systems wird in vergröberter Form durch Aggregate („Clouds") berücksichtigt. In diesem Sinne entwickeln wir ein Modell, welches jeweils einen Repräsentanten für Fileserver, Printserver und Windows-Terminal-Server enthält. Hinzu kommen die Netze und Netzkomponenten sowie vor allem der zentrale Catalyst-Switch.

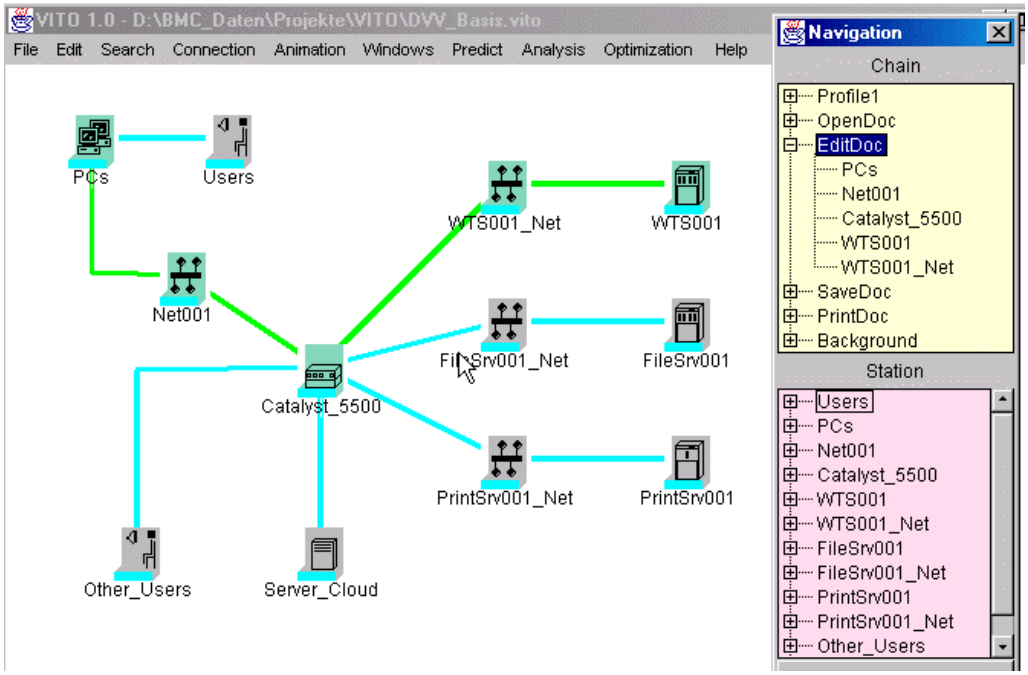


Abbildung 03-4: Layout des Basismodells, rechts das Navigationsfenster

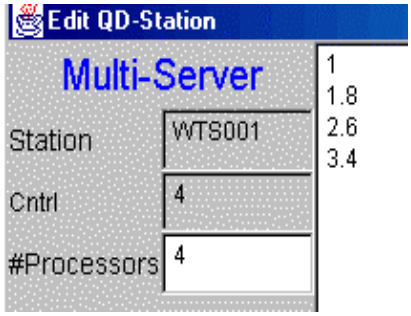


Abbildung 03-5: Skalierungsfaktoren für ein 4-Prozessorsystem

Edit Stations Help-F1

No	Station	Component	Strate...	Speed	Unit	Max....
1	Users	User	IS	-	-	-
2	PCs	PC-Group	IS	1.0	TA/sec	85.0
3	Net001	Fast-Ethernet	QD 40	100.0	Mbits/sec	40.0
4	Catalyst_5500	Switch	PS	10.0	Gbits/sec	50.0
5	WTS001	Multi-Server	QD 4	20.0	TA/sec	75.0
6	WTS001_Net	Fast-Ethernet	QD 40	100.0	Mbits/sec	40.0
7	FileSrv001	Multi-Server	QD 4	10.0	TA/sec	85.0
8	FileSrv001_Net	Fast-Ethernet	QD 40	100.0	Mbits/sec	40.0
9	PrintSrv001	Single-Server	PS	10.0	Jobs/sec	85.0
10	PrintSrv001_Net	Fast-Ethernet	QD 40	100.0	Mbits/sec	40.0
11	Other_Users	User	IS	-	-	-
12	Server_Cloud	Generic	PS	500.0	TA/sec	85.0

Abbildung 03-6: Liste der definierten Stationen

Die Abbildung 03-6 zeigt die Liste der 12 Stationen mit ihren Parametern. Der Windows-Terminal-Server WTS001 ist z.B. als sog. QD 4 Station (= 4-Prozessorsystem) modelliert; der Speed-Parameter mit Wert 20 TA/sec bezieht sich auf einen einzelnen Prozessor. Das Skalierungsverhaltenen wird durch sog. lastabhängige Bedienraten modelliert, vgl. hierzu Abbildung 03-5. Details zu dieser Modellierungstechnik finden sich in [VITO2000].

(c) Lastmodellierung

Intensität und Umfang der **Arbeitslasten** werden durch sogenannte Lastketten definiert, welche durch die Stationen des Modells führen, wozu auch Client-Groups zur Modellierung des Benutzerverhaltens gehören. Für die konkrete Modellierung der Arbeitslast ver-

wenden wir das Konzept der sogenannten Benutzerprofile, welche das Arbeitsverhalten und die daraus resultierende Ressourcenbelastung in hierarchischer Form beschreiben. Ein mögliches Profil ist z.B. die Ausführung von folgenden Transaktionen inklusive einer Denk- bzw. Pausenzeit:

1.	Öffnen eines Dokuments
2.	Editieren des Dokuments (zahlreiche Tastenanschläge und Mausbewegungen)
3.	Sichern des Dokuments (üblicherweise mehrfach inkl. Schließen des Dokuments)
4.	Drucken des Dokuments
5.	Denkzeit (bis zur erneuten Ausführung des Profils)

Tabelle 03-1: Benutzerprofil „Dokumentenbearbeitung“

Die einzelnen Transaktionen werden in VITO durch Ketten (engl. Chains) dargestellt, welche die Belastung von Netzen, Netzkomponenten, Servern und ggfs. auch Speichersystemen darstellen. Bei der Transaktion „Öffnen einer Datei“ haben wir Bearbeitungszeiten am PC, zunächst für die Bearbeitung des entsprechenden Kommandos und später den Aufwand für die Darstellung des Dokuments am Bildschirm, außerdem natürlich die Belastungen auf den Netzen und dem Switch zur Übertragung von Requests sowie vor allem für die Übertragung des Dokuments vom Fileserver zum Windows-Terminal-Server. Quittierungsverkehr und Protokolloverhead wird bei der Modellierung implizit als Bestandteil der gesamten Datenmenge betrachtet.

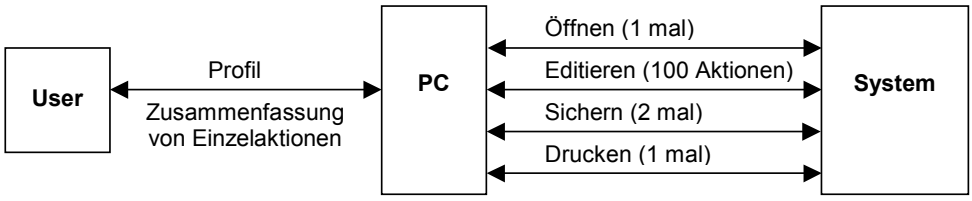


Abbildung 03-7: Hierarchische Lastbeschreibung[10]

Vom Windows-Terminal-Server zum Client-PC wird lediglich der Bildschirminhalt übertragen. In der Editierphase wird durch jede Eingabe per Tastatur oder Mausbewegung eine Interaktion mit dem Windows-Terminal-Server initiiert, wobei der Request vom Client

[10] Eine weitere Form der Darstellung, die zur Last- und Ablaufbeschreibung sehr hilfreich ist, sind die sog. Communications-Processing Delay Diagrams, vgl. [Mena1998].

zum WTS eine Größe < 100 Bytes besitzt, während in Gegenrichtung vom WTS zum Client jeweils Pakete mit einer mittleren Größe von einigen Hundert Bytes geschickt werden. Je nach konkretem Messszenario ergeben sich hier unterschiedliche Werte[11]. Für die Editieroperationen werden im Folgenden die mittleren Werte 100 bzw. 400 Byte (für In- bzw. OutPackets) verwendet.

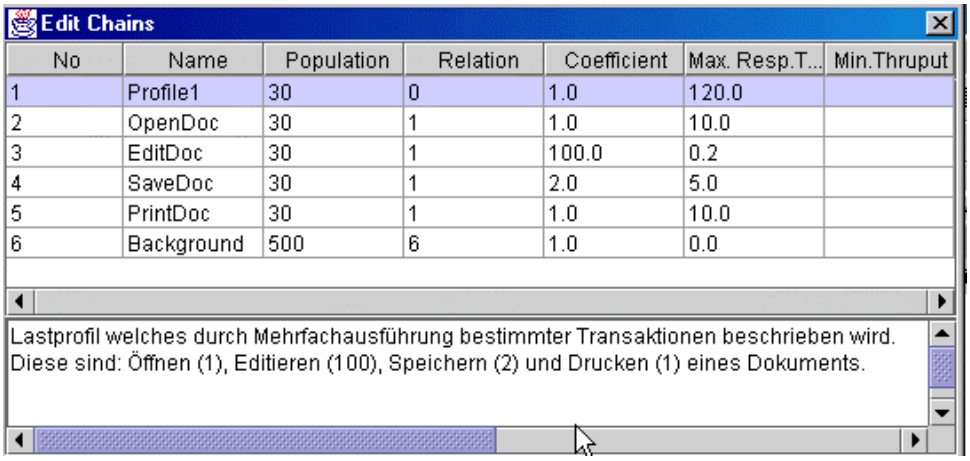


Abbildung 03-8: Lastkettendefinition in VITO

Abbildung 03-8 zeigt, wie das Benutzerprofil durch die Lastketten 2-6 verfeinert wird. Die Kette Profile1 ist hier auf einen Profildurchsatz von 1.0/sec eingestellt[12]. Die Ketten 2–6 sind bez. des Durchsatzes mit der Kette 1 gekoppelt und weisen jeweils ein Vielfaches des Durchsatzes von Kette 1 auf. Die Relation ist jeweils 1, d.h. die Durchsatzkopplung bezieht sich auf Kette 1. Entsprechend der hierarchischen Lastbeschreibung haben die Koeffizienten die Werte 1.0, 2.0 und 100, vgl. Abbildungen 03-7 und 03-8.

[11] Für die ICA-Clients werden für OutPackets mittlere Paketgrößen von 72-400 Bytes berichtet, für InPackets 300-1080 Bytes.

[12] Die ist ein etwas technischer Punkt: Ein „Coefficient 1.0“ in Verbindung mit „Relation 0“ bedeutet, dass durch sog. Kopplung mit „Kette 0“ ein Durchsatz von 1.0/sec eingestellt wird.

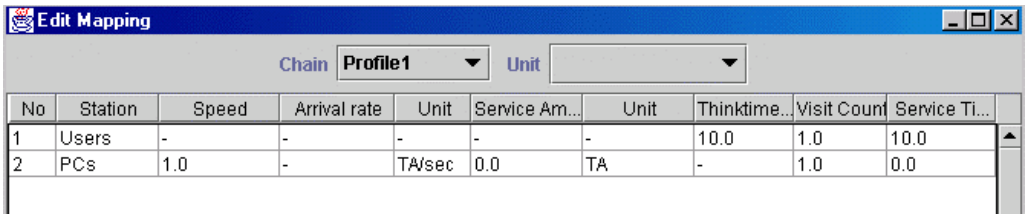

Edit Mapping
Chain Profile1 Unit
No Station Speed Arrival rate Unit Service Am... Unit Thinktime... Visit Count Service Ti...
1 Users - - - - - 10.0 1.0 10.0
2 PCs 1.0 - TA/sec 0.0 TA - 1.0 0.0

Abbildung 03-9: Mapping für das Benutzerprofil „Profile1“

Dieses Mapping legt fest, dass die Abarbeitung des Profils durch zyklische Interaktionen zwischen „Users“ und „PCs“ stattfindet. Ein Zyklus enthält 10 Sekunden Denk- bzw. Pausenzeit, was für die Station „Users“ als Thinktime mit dem Wert 10.0 eingetragen ist.

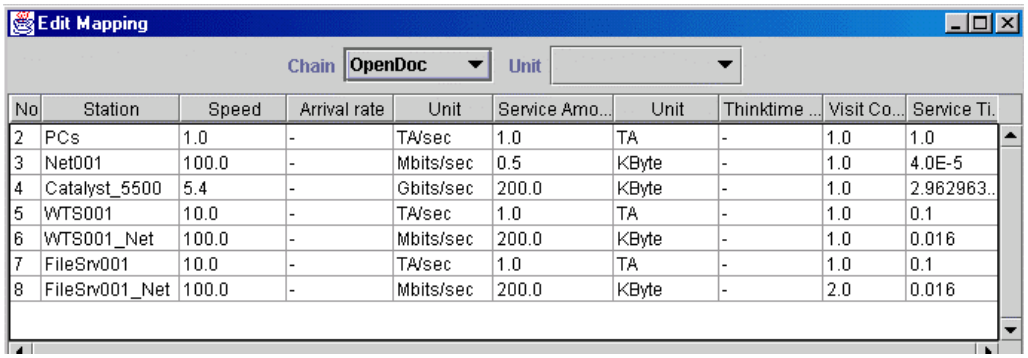

Edit Mapping
Chain OpenDoc Unit
No Station Speed Arrival rate Unit Service Amo... Unit Thinktime ... Visit Co... Service Ti.
2 PCs 1.0 - TA/sec 1.0 TA - 1.0 1.0
3 Net001 100.0 - Mbits/sec 0.5 KByte - 1.0 4.0E-5
4 Catalyst_5500 5.4 - Gbits/sec 200.0 KByte - 1.0 2.962963..
5 WTS001 10.0 - TA/sec 1.0 TA - 1.0 0.1
6 WTS001_Net 100.0 - Mbits/sec 200.0 KByte - 1.0 0.016
7 FileSrv001 10.0 - TA/sec 1.0 TA - 1.0 0.1
8 FileSrv001_Net 100.0 - Mbits/sec 200.0 KByte - 2.0 0.016

Abbildung 03-10: Mapping für Öffnen eines Dokuments „OpenDoc“

Eine der Transaktionen des Profils ist das Öffnen des Dokuments (OpenDoc). Die entsprechenden Ressourcenverbräuche (plausible Schätzwerte) sind in dem o.g. Dialog eingetragen, vgl. Abbildung 03-10. Aus „Service Amount“ und „Speed“ sowie ggf. unter Verwendung der „Unit“ wird die „Service Time“ errechnet.

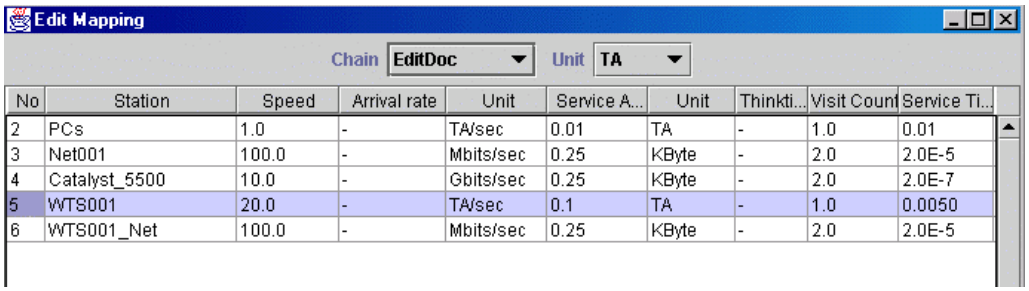

Edit Mapping
Chain EditDoc Unit TA
No Station Speed Arrival rate Unit Service A... Unit Thinkti... Visit Count Service Ti...
2 PCs 1.0 - TA/sec 0.01 TA - 1.0 0.01
3 Net001 100.0 - Mbits/sec 0.25 KByte - 2.0 2.0E-5
4 Catalyst_5500 10.0 - Gbits/sec 0.25 KByte - 2.0 2.0E-7
5 WTS001 20.0 - TA/sec 0.1 TA - 1.0 0.0050
6 WTS001_Net 100.0 - Mbits/sec 0.25 KByte - 2.0 2.0E-5

Abbildung 03-11: Mapping für das Bearbeiten eines Dokuments „EditDoc“

Eine weitere Transaktion des Profils ist das Editieren des Dokuments (EditDoc). Hier haben wir relativ geringe Ressourcenverbräuche, die aber häufig angefordert werden, und zwar 100 mal pro Abarbeitung des Profils. Hier ist für eine „leichte“ Editieraktion (Zeile 4) der Service Amount nur mit 0.1 TA angegeben. Die Mappings für Sichern (SaveDoc) und Drucken (PrintDoc) sind entsprechend aufgebaut und hier nicht dargestellt. Um den File Server FileSrv001 mit in die Modellierung einzubeziehen, wird die sog. Background-Load verwendet, welche durch Öffnen und Sichern von Dokumenten jeweils Belastungen am File Server hervorruft. Abbildung 03-12 zeigt die Visualisierung dieser Lastkette.

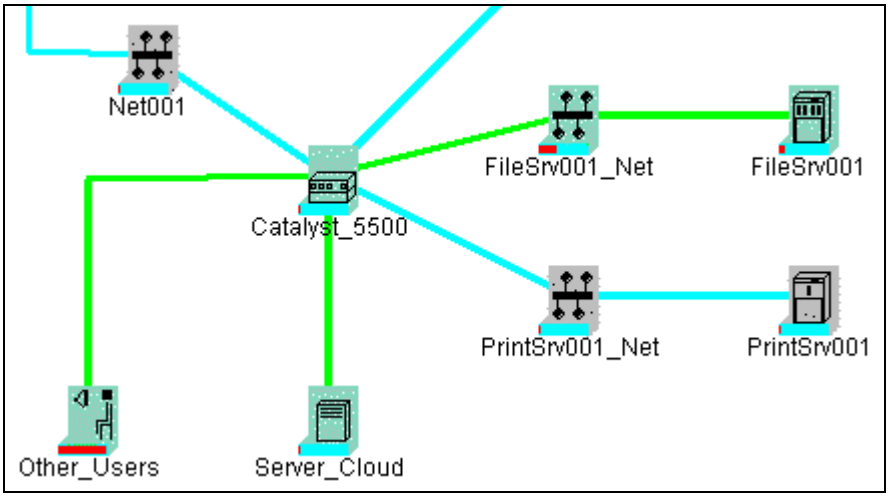


Abbildung 03-12: Die Lastkette „Background“

03.08 Szenario: IST-Situation Windows-Terminal-Server

Die Auswertung des in den vorstehenden Abschnitten dargestellten Szenarios führt zu folgenden Ergebnissen. Abbildung 03-13 zeigt z.B. die Antwortzeiten („Response Times“).

Results

No	Chain	Popul.	Relat.	Coeff.	Thruput	Forced Dels	Response
1	Profile1	30	0	1.0	1.0	20.0	0.00000
2	OpenDoc	30	1	1.0	1.0	28.53032	1.46967
3	EditDoc	30	1	100.0	100.0	0.28316	0.01683
4	SaveDoc	30	1	2.0	2.0	13.61082	1.38918
5	PrintDoc	30	1	1.0	1.0	28.75518	1.24481
6	Background	500	6	1.0	4.9780	0.00000	0.44078

Abbildung 03-13: Antwortzeiten für die Lastketten

Die Response Times für die Ketten OpenDoc, EditDoc, SaveDoc und PrintDoc genügen den vereinbarten Dienstgüten.

No	Station	Component	Typ	Speed	Unit	Populatio	Util [%]	UtilUsr [	UtilOhd	Idle [%]
1	Users	User	IS	------	------	------	------	------	------	------
2	PCs	PC-Group	IS	1.0	TA/sec	5.0	4.166	------	------	95.833
3	Net001	Fast-Ethernet	QD 40	100.0	Mbits/se	0.0145	1.412	------	------	98.588
4	Catalyst_5500	Switch	PS	10.0	Gbits/se	0.0009	0.092	------	------	99.907
5	WTS001	Multi-Server	QD 4	20.0	TA/sec	1.2151	24.314	22.5	1.814	75.685
6	WTS001_Net	Fast-Ethernet	QD 40	100.0	Mbits/se	0.1166	9.2	------	------	90.8
7	FileSrv001	Multi-Server	QD 4	10.0	TA/sec	2.0125	34.106	32.390	1.715	65.893
8	FileSrv001_Net	Fast-Ethernet	QD 40	100.0	Mbits/se	0.7937	27.912	------	------	72.087
9	PrintSrv001	Single-Server	PS	10.0	Jobs/sec	0.1107	10.0	------	------	90.0
10	PrintSrv001_Net	Fast-Ethernet	QD 40	100.0	Mbits/se	0.0253	2.4	------	------	97.6
11	Other_Users	User	IS	------	------	------	------	------	------	------
12	Server_Cloud	Generic	PS	1000.0	TA/sec	0.0806	7.467	------	------	92.532

Abbildung 03-14: Station View – Ergebnisse für Stationen

Wir sehen, dass der Windows-Terminal-Server WTS001 eine Auslastung von 24,3 % hat, wovon 22,5 % auf Benutzeraufträge (UtilUsr) und 1,8 % auf System Overhead (UtilOhd) entfallen. Dieses Ergebnis bezieht sich auf die vorausgesetzte Skalierung des WTS001, die als Modelleigenschaft im VITO-Modell durch entsprechende Parameterwerte eingetragen ist. Bei höheren Auslastungen wird der System Overhead überproportional ansteigen. Die Auslastung des FileSrv001, der vor allem von der Lastkette „Background“ belastet wird, ist hier mit 34.1 % berechnet.

03.09 Szenario: Performance Prognose bei Erhöhung der Arbeitslast

Die Betrachtung der System-Performance unter erhöhter Arbeitslast geschieht durch Variation des „Coefficient“ für die Kette „Profile1“ im Wertebereich 1.0-4.0. VITO stellt hierfür eine Option zum interaktiven Modellieren bereitet. Die erzeugten Grafiken zeigen, das sich bei der gewählten Modellierungsmethodik ein Grenzdurchsatz von ca. 3.35 Profilen pro Sekunde einstellt. Dies entspricht einem Durchsatz von ca. 335 elementaren Editieroperationen am WTS.

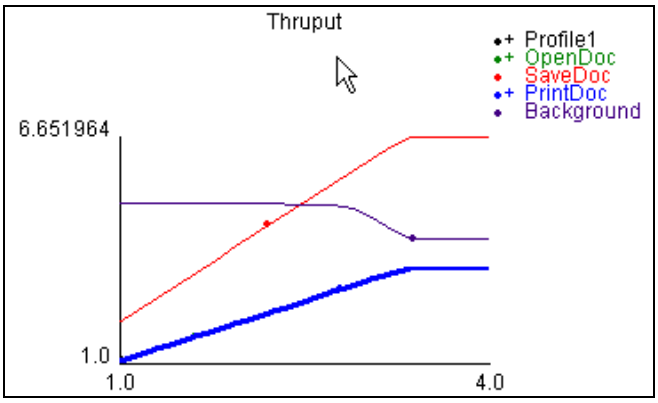


Abbildung 03-15: Ketten-Durchsätze für 1.0 bis 4.0 Profile pro Sekunde

Die Durchsätze für Profile, OpenDoc und PrintDoc sind identisch. SaveDoc ist doppelt so hoch. Die Background-Kette bleibt zunächst unverändert, wird aber im Hochlastbereich gebremst. Der maximal erzielbare Durchsatz ist 3.35 Profile pro Sekunde.

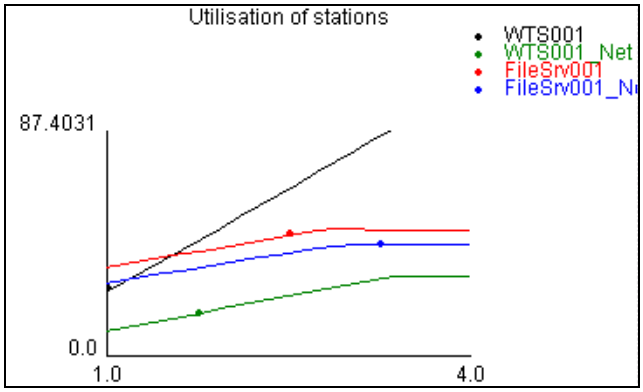


Abbildung 03-16: Server- und Netzauslastungen

Betrachtet man das Auslastungsverhalten, Abbildung 03-16, so kann man vermuten, dass der Server WTS001 der Engpass ist. Passend dazu das Antwortzeitverhalten, vgl. Abbildung 03-17.

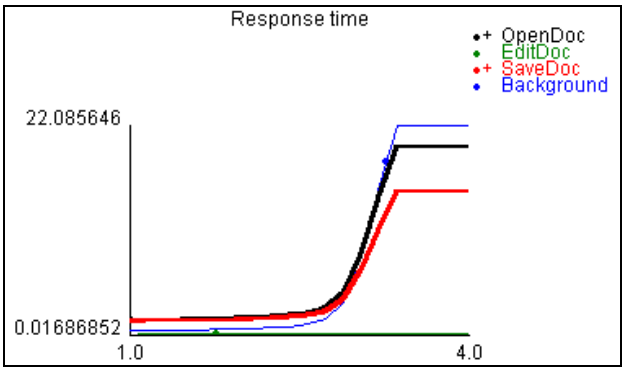


Abbildung 03-17: Exponentieller Anstieg der Antwortzeiten ab Durchsatz 2.8 Profile/sec

In Abbildung 03-17 sieht man, dass ab 3.35 Profile/sec eine Saturierung einsetzt. VITO „friert“ den Durchsatz auf diesem Niveau ein.

03.10 Ausblick

Diese Arbeit erhebt nicht den Anspruch, ein völlig realistisches Modell eines existierenden IT-Systems zu sein. Zur Erstellung eines konkreten Basismodells sind zusätzlich zu den Messungen auf Paketebene auch Messungen der Ressourcenverbräuche an den Servern (Auslastungen, Durchsätze) notwendig. Im Idealfall wird auch eine Antwortzeitüberwachung und eine grobe Klassifizierung des Benutzerverhaltens im Sinne von Benutzerprofilen durchgeführt.

Unter diesen Voraussetzungen kann VITO dazu beitragen, ein systematisches Kapazitätsmanagement aufzubauen.

03.11 Literatur

[Jain1991] R. Jain; The Art of Computer System Performance: Analysis, Techniques for Experimental Design, Measurement, Simulation and Modeling, Wiley, 1991

[Mena1998] D.A. Menasce, V.A.F. Almeida; Capacity Planning for Web Performance: Metrics, Models, and Methods, Prentice Hall, 1998

[Mena2000] D.A. Menasce, V.A.F. Almeida; Scaling for e-Business, Prentice Hall, 2000

[VITO2000] B. Müller-Clostermann (ed.); Capacity Planning and Performance Evaluation with the Tool VITO, Informatik, Universität Essen, 2000

KAPITEL 04 DAS WERKZEUG ECOPREDICTOR

TORSTEN BERNERT, BRUNO MÜLLER-CLOSTERMANN

04.01 Einführung: Analytische und simulative Modelle

Zunächst soll das Netzplanungswerkzeug EcoPredictor[13] abgegrenzt werden gegenüber simulativ arbeitenden Programmen, die für diesen Zweck ebenfalls prinzipiell auch zum Einsatz kommen können. Da die Funktionsweise auf der analytischen Berechnung von Warteschlangennetzen beruht, ist der EcoPredictor für die Netzplanung besser geeignet als simulative Werkzeuge. Predictor ist also kein Werkzeug, welches konkrete Simulationsereignisse erzeugt (z.B. Generierung von Paketen) und die jedem Ereignis zugeordnete Ereignisroutine, z.B. Zeitverbrauch im lokalen Netz und ggf. Kollisionsbehandlung, dann abarbeitet.

Da komplexe und oft heterogene Netze und ihre Komponenten wie Server, Switches usw. in einer Sekunde Tausende bis Millionen von Paketen bewältigen, würde jedes Paket während einer Simulation mehrere Ereignisse auslösen (z.B. Ankünfte und Abgänge an/von Komponenten, verlorengegangene Nachrichten etc.), die vom Simulator behandelt werden müssen. Auf dieser detaillierten Ebene kann eine simulierte Realzeitstunde Tage oder gar Wochen in Anspruch nehmen. „Testweise“ ein weiteres LAN-Segment einzufügen oder durch eine neue Anwendung hervorgerufene Paketflüsse hinzuzufügen wird zu hohen Aufwänden führen, die in der Regel nicht tragbar sind. Die Komplexität des Modells verhindert zudem seine Pflege, d.h. der Aufwand, das Modell ständig mit dem realen Netzwerk abzugleichen, verursacht Kosten, die den entstehenden Nutzen bei weitem übersteigen.

Tools wie der Predictor dagegen ermöglichen es, aufgrund ihres analytischen Lösungsansatzes auch Modelle umfangreicher Netzwerkarchitekturen in wenigen Sekunden zu lösen. Die zugrunde liegende Theorie der Warteschlangennetze kommt zudem mit wenigen Eingangsdaten aus; im Wesentlichen sind dies: Ankunfts- und Bedienraten, Bedienstrategien, Stationstypen, Informationen über das Routingverhalten, z.B. in Form von Wechselwahrscheinlichkeiten oder Besuchshäufigkeiten, offenes oder geschlossenes Netz, sowie ggf. Angaben über Variabilität von Paketraten und Paketgrößen. Dies erleichtert somit i.d.R. die Parametrisierung des Modells. Die Berechnung des Modells erledigt der Predictor mit Hilfe

[13] Das Werkzeug EcoPredictor wurde bis zum Jahr 2000 unter dem Namen COMNET Predictor vertrieben.

eines Dekompositionsverfahrens, das approximative Lösungen liefert. Auch hier müssen u.U. komplexe Gleichungen gelöst werden, doch entfällt ein simulatives, imitierendes Fortschreiten entlang der Zeitachse, da die Dauer des Beobachtungsintervalls nicht in die analytischen Berechnungen eingeht und stattdessen stationäre Analysen mit Mittelwertberechnungen durchgeführt werden. Daher sind diese Verfahren auch bei komplexen Problemstellungen, bei denen ein Simulator auf viele Variablen und Ereignisse Rücksicht nehmen muss, sehr schnell.

Die gegenüber einem simulativen Modell begrenzte Parameteranzahl und die hinsichtlich einiger Ergebnisgrößen approximativen Lösungen könnten nun als Nachteil gewertet werden. Man muss sich jedoch vor Augen halten, dass der Predictor für die konzeptuelle Planung von Netzwerken, für das Erkennen von Trends und Leistungsengpässen und nicht zuletzt auch für die Dimensionierung zukünftiger Systeme unter vorherzusagenden Netzlasten eingesetzt wird. Bei der letztgenannten Aufgabe muss ohnehin schon mit unbekannten Prognosewerten bei der Parametrisierung gearbeitet werden, denn kein Unternehmen wird seinen künftig erwarteten Netzverkehr sehr exakt beschreiben können. Generell wird bei diesen Planungsaufgaben keine Genauigkeit bis auf die Nachkommastelle erwartet. Durch die Algorithmen bedingte Approximationen mit Abweichungen von 5 bis 10 Prozent sind in diesem Kontext tolerierbar. Ein kompakter Parameterssatz erleichtert weiterhin die Erhebung der Daten, die Modellpflege sowie das Erstellen von bzw. das Experimentieren mit Zukunftsszenarien.

Im Folgenden wird die Modellierung mit dem Predictor vorgestellt. Der erste Abschnitt erläutert überblicksartig den Ablauf, die folgenden Abschnitte vertiefen die einzelnen Schritte und binden sie in ein Vorgehensmodell ein.

04.02 Modellerstellung mit Predictor

Die Erstellung eines ersten Modells in Predictor ist prinzipiell einfach und intuitiv durchführbar. In Drag-and-Drop-Manier werden kleine Symbole, welche verschiedene Netzwerkkomponenten repräsentieren, aus einer Icon-Leiste in das Predictor-Fenster gezogen und dort zu einem Netzwerk verbunden. Die Komponenten lassen sich hinsichtlich ihrer Leistungsfähigkeit (z.B. in *Paketen/sec*) parametrisieren. Anschließend sind in einer Tabelle die Lasten – in Predictor heißen diese *Packet Flows* – per Quelle-Ziel-Beziehung zu definieren. Die Ankunftsrate in *Packets/sec* und die durchschnittliche Größe eines Pakets in *Avg Bytes/Pkt* sind die beiden wesentlichen Parameter, welche pro *Packet Flow* eingetragen werden. Ein Tastendruck auf F4 führt eine Konsistenzprüfung durch (Gibt es isolierte Komponenten? Sind alle in der Tabelle aufgeführten Quellen und Ziele noch im Modell vorhanden? … oder wurden sie zwischenzeitlich gelöscht? ...) und startet dann die „Berechnung der Performance“. Nach ca. einer Sekunde Rechenzeit stehen tabellarische und

grafische Reports zur Verfügung, die sich nach verschiedenen Zielgrößen sortieren lassen, oder die selektierbaren Leistungsmaße in Form von Balkendiagrammen präsentieren.[14]

04.03 Systematische Kapazitätsplanung mit dem Predictor

Wenn Forschungsergebnisse Einzug in die Praxis halten und bisher übliche Denkweisen und Abläufe ergänzen oder ablösen, werden Empfehlungen zur Systematisierung der neuen Vorgehensweise unterbreitet. Arbeitsgänge, die sich als günstig herausgestellt haben, werden als eine Art Leitfaden präsentiert. Derartige Vorgehensmodelle haben mittlerweile auch in der Kapazitätsplanung Fuß gefasst und sind in diversen Varianten vertreten. Auf den ersten Blick recht verschieden, ähneln sie sich nach genauerem Hinschauen doch recht stark, sind meist nur mehr oder weniger detailliert ([Mena94] vs. [HaZo95]), beziehen auch vorgelagerte Schritte mit ein [MAP2000] oder sind auf die Kapazitätsplanungssoftware einer Firma hin zugeschnitten [MakeSys98]. Die Schritte eines Vorgehensmodells sollen als Bezugpunkte während der Modellierungsphasen dienen. Daniel Menascé, an dessen Vorgehensmodell wir uns im Folgenden orientieren, schlägt nachstehende Schritte während des Kapazitätsplanungsprozesses vor:

- Verstehen des Systems[15] der Hardware- und Softwareumgebung („IST-Analyse“)
- Charakterisierung der Lasten in Form von Packet Flows
- Erstellen eines Systemmodells
- Validierung des Modells
- Prognostizieren der zukünftigen Lasten („Lastprognose“)
- Vorhersage des zukünftigen Leistungsverhaltens („Performance Prognose“)

Schritt 1 ist klar: Es kann nichts modelliert werden, ohne das der Aufbau und die prinzipielle Funktionsweise des „Untersuchungsobjekts“ verstanden ist. Egal, ob ein existierendes oder ein neu zu planendes System in ein Modell überführt werden soll, seine grobe Struktur und seine Basiskomponenten müssen vor der Modellierung feststehen. Änderungen und Ergänzungen lassen sich ab Schritt 4 einbringen. Wenn ein Netzwerk-Management-System die Topologieinformationen eines existierenden Systems exportieren kann, dann lassen sich diese Informationen über den sog. *Baseliner* importieren; Anzahl und Beziehungen der Komponenten untereinander sind anschließend im Modell präsent.

[14] Einige Reports sind in Abschnitt 04.04 zu finden.

[15] Wenn im Folgenden von „System“ die Rede ist, so ist immer ein System bestehend aus Hard- und Software und Applikationen gemeint.

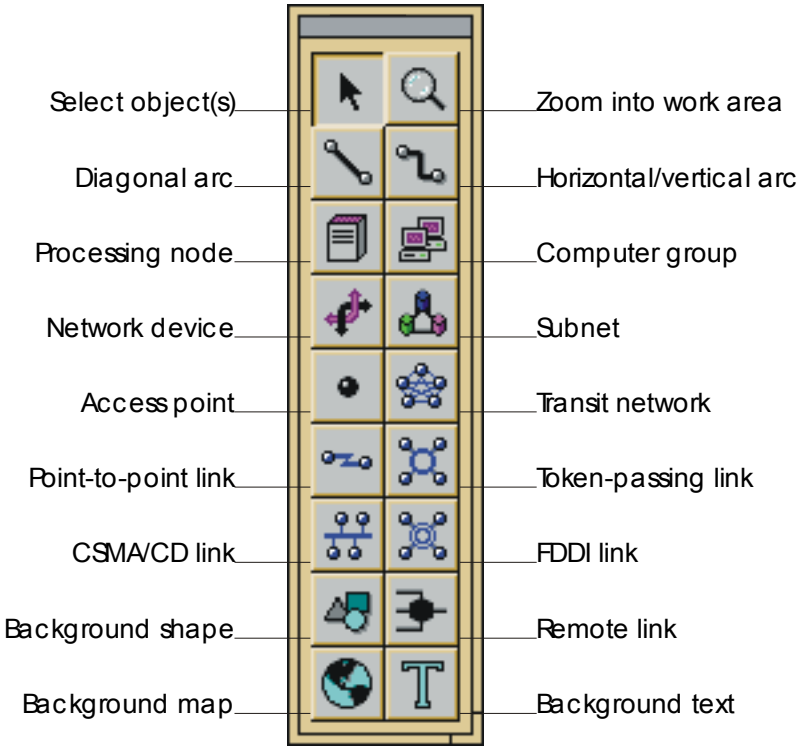


Abbildung 04-1: Komponenten im Predictor

Ansonsten bietet der Predictor für die Umsetzung eines Netzwerkes in ein Modell einen Fundus an typischen Netzkomponenten, die initial mit Defaultwerten belegt sind (vgl. Abbildung 04-1). Ihre Kapazitäten lassen sich an die der realen Komponenten anpassen. Dabei kann im Falle der Links und Network Devices oft aus einer Bibliothek die entsprechende Hardware ausgewählt werden. Generell werden Links im Wesentlichen durch ihre Bandbreiten und minimalen/maximalen Frame-Längen beschrieben. Network Devices ebenfalls durch ihre Bandbreite oder ihre Kapazität in packets-per-second. Processing Nodes lassen lediglich eine Dimensionierung in packets-per-second zu. Alle diese Komponenten lassen sich aber mit verschiedenen Protokollen nutzen, und ihre Kapazitäten können je nach Protokoll differieren.

Um große Netze hierarchisch zu strukturieren, können sogenannte Transit Networks oder Subnets angelegt werden. Logisch oder physikalisch zusammengehörige Teilnetze lassen

sich in einer solchen Komponente zusammenfassen und liegen dann im Modell hierarchisch gesehen eine Stufe tiefer. Eine derartige Schachtelung kann beliebig tief sein.

Schritt 2 des Vorgehensmodells sieht eine Charakterisierung der Arbeitslasten vor. In Netzwerken ist hier in der Regel von Paketen die Rede. Eine weitere Differenzierung des Begriffs „Paket“ in verschiedenen Netztypen (Ethernet, ATM, Token Ring, ...) oder gar auf verschiedenen Schichten des ISO/OSI-Modells findet an dieser Stelle nicht statt.

Vor allem bei großen Netzen ist es sinnvoll, mit Hilfe von Netzwerk-Analyzern oder Monitoring-Werkzeugen Verkehrs- und Topologiedaten eines bestehenden Netz zu erfassen und zu exportieren. Sie lassen sich in den Predictor importieren und beschreiben die Verkehrsmatrix bzw. die Verkehrsflüsse, die im Netz auftreten.

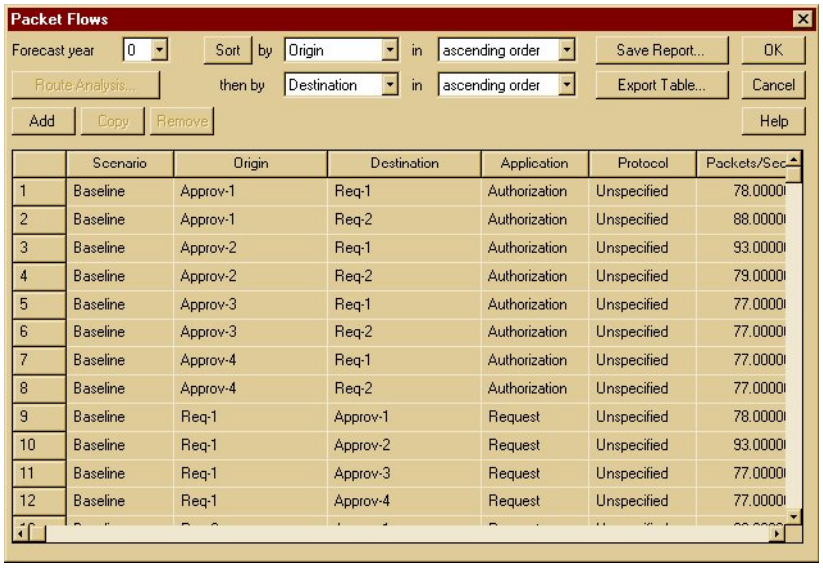


Abbildung 04-2: Spezifikation von Packet Flows

Oft müssen die Lasten jedoch „per Hand“ eingegeben werden. Im Predictor geschieht dies durch die Beschreibung sogenannter Packet Flows (vgl. Abbildung 04-2), wobei zunächst Quelle-und Ziel (Origin, Destination) anzugeben sind. Die Ankunftsrate ist in *Packets/Sec* und die mittlere Größe eines Pakets in *Avg Bytes/Pkt* anzugeben. Folgende Parameter „verfeinern“ diese Beschreibung:

@ Dem Packet Flow lassen sich ein bestimmtes *Scenario* (z.B. „Ist-Situation“), eine *Application* (z.B. „DB-Anfrage“) und sein *Protocol* (z.B. „IP“) zuordnen.

@ Der Parameter *Burstiness* definiert die Variabilität des Paketankünfte. Ein Wert von Null beschreibt einen deterministischen Ankunftsstrom, die Pakete kommen in regelmäßigen Abständen („getaktet"). Belässt man den Wert auf „1", so genügen die Ankünfte den Voraussetzungen, wie sie vom M/M/1-Modell bzw. offenen Warteschlangennetzen her bekannt sind (Poisson-verteilt). Je höher der Wert gesetzt wird, desto größer ist die Variabilität des Ankunftsstroms, d.h. die Ankünfte der Packet Flows zeigen ein burst-artiges Verhalten. Ankünfte treten beispielsweise nach Ruhepausen vermehrt in Gruppen auf.

@ Die *StDev Pkt Size* ist die Standardabweichung der Paketgröße in Bytes. Der Default-Wert Null definiert eine konstante Paketgröße.

Nach diesen Beschreibungen wird klar, dass der Predictor für die interne Repräsentation der Netzwerkmodelle offene Warteschlangennetze mit Stationen vom Typ G/G/1 bzw. G/G/m verwendet.

Schritt 3 könnte beinahe übersprungen werden, denn das Modell des Systems wurde ja bereits erstellt. Dennoch gibt es etwas zu tun: Zwar wurden Quelle und Ziel eines jeden Packet Flows definiert, doch wie findet ein Flow seinen Weg durchs Netzwerk? Was ist, wenn es mehrere Wege gibt? – Wechselwahrscheinlichkeiten zwischen Stationen bzw. Besuchshäufigkeiten werden nicht angegeben, statt dessen stehen im Predictor verschiedene Routing-Protokolle zur Verfügung, die den Weg der Packet Flows durchs Netzwerk bestimmen.

Jeder Link hat einen Parameter *Routing Metrics*, der eine Art Gewichtung dieses Links darstellt. Standardmäßig sind hier alle Links mit „1" gewichtet. Dies lässt sich jedoch für jedes zulässige Übertragungsprotokoll (IP, X.25, ...) separat ändern. Das voreingestellte Routing-Protokoll „RIP Minimum Hop" findet den kürzesten Weg für einen Packet Flow, indem alle möglichen Wege mit Hilfe der Summenbildung der Link-Gewichte bewertet werden und die günstigste Alternative ausgewählt wird. Stehen mehrere Wege mit gleich niedriger Summe zur Verfügung, so erfolgt die Auswahl zufällig. Es existieren weitere Protokolle, welche die Flows gleichzeitig auf verschiedene Wege routen können, so dass betroffene Komponenten möglichst gleichmäßig oder nicht zu hoch ausgelastet werden. Das aufwändigste „IGRP"-Routing bezieht gar die Bandbreite und die Netzverzögerung eines Links anteilsmäßig in seine Routingstrategie mit ein.

Steht das Routingverfahren fest, so sind auch die besuchten Komponenten eines jeden Packet Flows bekannt und können in der später besprochenen analytischen Berechnung des Modells entsprechend berücksichtigt werden.

Schritt 4 beschäftigt sich mit der Validierung des Modells. Dieser Schritt kann eigentlich nur in befriedigender Weise durchgeführt werden, wenn in einem existierenden System Messwerte erhoben werden können, mit denen das Modell bez. seiner Prognosegüte abge-

glichen werden kann. An dieser Stelle lassen sich auch Fragen beantworten, wie z.B. „Was passiert, wenn Netzwerkkomponenten ausfallen?“ oder „Hilft redundant ausgelegte Hardware die kritischen Dienste unter sanftem Leistungsabfall aufrechtzuerhalten?“ Um dies zu untersuchen, lässt sich an den entsprechenden Komponenten der Parameter *Failed* einschalten. Eine solche Komponente gilt als ausgefallen und die Routing-Algorithmen versuchen nun, für die betroffenen Packet Flows andere Wege durchs Netzwerk zu finden.

Auf eine Vertiefung des Themas Kalibrierung/Validierung soll an dieser Stelle verzichtet werden, da der Predictor in diesem Zusammenhang keine weitere Unterstützung bietet. Einzelheiten lassen sich z.B. in [Mena94, Kapitel 10] nachlesen.

Schritt 5 ist eine der Stärken des Predictor, wie die Namensgebung schon vermuten lässt. So heißt es gleich in der Einführung des Getting Started Guide: „COMNET Predictor is designed for network managers and planners who need to predict how changes will affect their networks ...“. Diese Änderungen können sich natürlich einerseits auf eine Änderung im Netzwerk selbst beziehen, d.h. von der Parameteränderung einzelner Komponenten bis hin zur Neugestaltung ganzer Teilnetze lassen sich vielfältige Optionen erproben. Andererseits ist aber sicherlich auch eine Änderung in der Last zukünftiger Perioden gemeint. In den sogenannten *Forecast Parameters* lassen sich Art und Anzahl der Vorhersageperioden eintragen. Wie sehr der Predictor hier seinem Namen verpflichtet ist, merkt man daran, dass er mindestens immer eine Periode vorhersagt; ganz ausschalten lässt sich dieser Forecast-Mechanismus nicht. Perioden können Stunden, Tage, ..., Quartale oder Jahre sein.

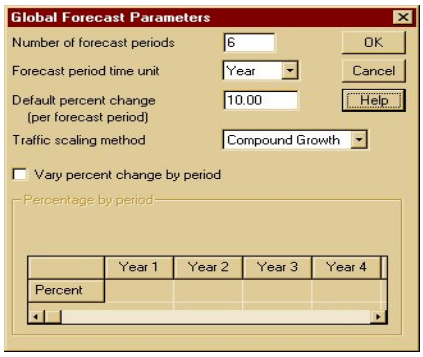


Abbildung 04-3: Angabe von Forecast Parametern

Die einfachste Art, den Umfang der Laststeigerung im Predictor einzustellen, ist eine globale prozentuale Lastanhebung. Mit den Einstellungen in der folgenden Abbildung würde der Predictor die Leistungsmaße über einen Zeitraum von 6 Jahren (= 6 Perioden) mit einer jährlichen Laststeigerung von 10 Prozent berechnen. Jede Periode ist in sämtlichen Reports

– diese werden im nächsten Schritt des Vorgehensmodells besprochen – vertreten. Globale Laständerungen, die pro Periode variieren, lassen sich ebenso realisieren, wie detaillierte Laständerungen getrennt für Applikationen, Übertragungsprotokolle und Szenarios. Das Werkzeug unterstützt den Netzplaner also beim Einstellen der in zukünftigen Perioden erwarteten Lasten, kann die Laststeigerung selbst aber nicht vorhersagen. Hier muss man auf Trend- oder Regressionsanalysen zurückgreifen. Auch Schätzungen aufgrund von Erfahrungswerten oder von erwarteten Zuwächsen im Lastaufkommen aufgrund neuer oder geänderter Geschäftsprozesse können hilfreich sein.

Die Vorhersage des zukünftigen Leistungsverhaltens des Systems realisiert den letzten, **Schritt 6** des Vorgehensmodells. Zu diesem Zweck generiert der Predictor eine Vielzahl an tabellarischen und grafischen Reports, die diverse Leistungsmaße nach verschiedenen Kriterien aufbereiten.

Die Ergebnisse einer Modellrechnung können nur so gut sein, wie das Modell und seine Paramter selbst. Zwar können Teile des Netzwerks im Modell weggelassen oder stark abstrahiert werden, wenn jedoch Aussagen zum Verhalten dieses Netzes zu machen sind, so dürfen diese Vereinfachungen sich nicht auf den betrachteten Bereich erstrecken oder ihre Auswirkungen müssen hinreichend genau verstanden werden, um abweichende Ergebnisse gedanklich zu kompensieren. Zusammenfassend gilt wie für alle Werkzeuge das sog GI-GO-Prinzip: „Garbage-In-Garbage-Out“ (Eingabedaten=Müll → Ausgabedaten = Müll)!

04.04 Beispiel für ein Predictor-Modell

Ein Beispielmodell wurde für drei Jahre berechnet: Für das aktuelle Jahr und zwei Folgejahre wird ein globaler Lastzuwachs von 10 Prozent angenommen. Als Ergebnis sind im Jahr 0 drei Warnungen zu verzeichnen, in den beiden folgenden wird gar ein Alarm angezeigt. Ab wann berechnete Leistungsmaße Warnungen und Alarme auslösen, lässt sich in den *Warning and Alarm Thresholds* für Network Devices, Processing Nodes und alle Links getrennt angeben.

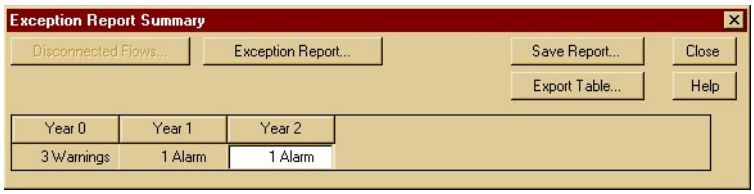


Abbildung 04-4: Warnungen und Alarme

Die *Network Resource Analysis* ist ein Report, der einen Überblick über Leistungsmaße wie Auslastung, Warteschlangenlänge oder Wartezeiten in einzelnen Netz-Komponenten liefert.

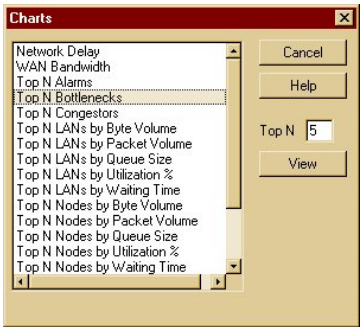

Charts
Network Delay
WAN Bandwidth
Top N Alarms
Top N Bottlenecks
Top N Congestors
Top N LANs by Byte Volume
Top N LANs by Packet Volume
Top N LANs by Queue Size
Top N LANs by Utilization %
Top N LANs by Waiting Time
Top N Nodes by Byte Volume
Top N Nodes by Packet Volume
Top N Nodes by Queue Size
Top N Nodes by Utilization %
Top N Nodes by Waiting Time
Cancel
Help
Top N 5
View

Abbildung 04-5: Auswahl eines Leistungsmaßes

	Nodes and Links	Year 0	Year 1	Year 2
1	Frame Relay.T1 Line	75.37	82.90	91.19
2	Frame Relay.CIR18-12	60.20	66.22	72.84
3	Frame Relay.Access A2	60.20	66.22	72.84
4	Frame Relay.Access A1	58.10	63.91	70.30
5	Frame Relay.CIR17-18	58.10	63.91	70.30
6	Frame Relay.Access R1	56.88	62.56	68.82
7	Frame Relay.CIR10-18	56.88	62.56	68.82
8	Frame Relay.Access R2	56.18	61.79	67.97
9	Frame Relay.CIR11-18	56.18	61.79	67.97
10	Frame Relay.CIR23-18	53.90	59.29	65.22
11	Frame Relay.Access A3	53.90	59.29	65.22

Abbildung 04-6: Auswahl von Komponenten

Weiterhin steht eine Vielzahl von *Charts* zur Verfügung, welche Leistungsmaße oder eine andere Zielgröße (z.B. „Bottleneck“) grafisch aufbereitet. Hier lassen sich die Auswirkungen von Laststeigerungen über die Perioden besonders gut beobachten.

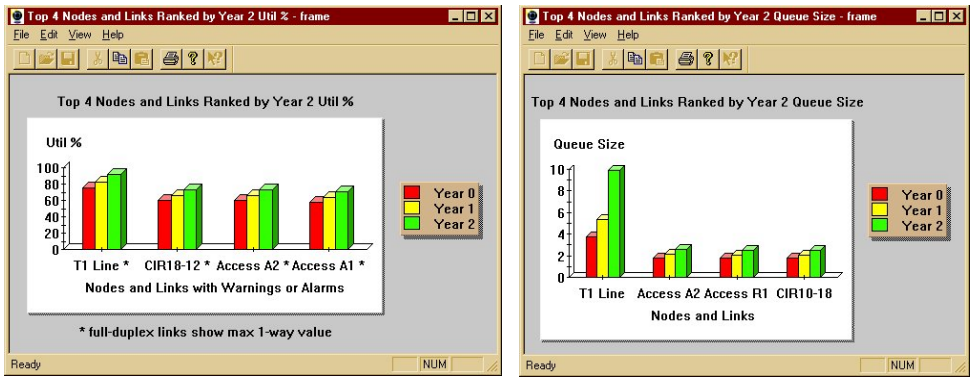


Abbildung 04-7: Diagramme zu Auslastung und Warteschlangenlänge

04.05 Zur Analysetechnik des EcoPREDICTOR

Das abschließende Kapitel beleuchtet etwas die Grundbegriffe aus der Warteschlangentheorie, die im Predictor von Bedeutung sind. Verständlicherweise sind in der Dokumentation nicht die Algorithmen offengelegt, aber einige Anhaltspunkte lassen sich finden.

Die Grundlagen der im Predictor verwendeten Analysemethode wurden in einem „White Paper“ der Firma CACI veröffentlicht, welche den Predictor ürsprünglich entwickelt und distributiert hat. Als erstes wird für jeden Packet Flow auf der Grundlage des eingestellten Routing-Verfahrens der kürzeste bzw. günstigste Weg ermittelt. Auf dieser Basis wird die Topologie des Netzwerks in ein Wartesschlangennetz überführt. So wird beispielsweise jedes Network Device auf ein G/G/1-Wartesystem und jeder point-to-point WAN link auf ein Paar von G/G/m-Wartesystemen abgebildet.

Der erste Parameter der Kendall-Notation G/G/m steht für die Verteilung der Zwischenankunftsabstände – hier können diese beliebig verteilt sein (G=General, allgemein). Im konkreten Fall werden Verteilungen durch Mittelwert und Variationskoeffizient beschrieben. Beim Predictor sind dies die Paketrate (= 1/Mittelwert) und Burstiness (= Variationskoeffizient). Der zweite Parameter beschreibt die Verteilung der Bedienzeiten – auch diese ist hier beliebig. Im Predictor kann neben der mittleren Paketgröße deren Standardabweichung angegeben werden. Der dritte Parameter kennzeichnet die Anzahl der Bediener. Bei G/G/1 ist ein Bediener, bei G/G/m sind m Bediener vorhanden.

Das Warteschlangennetzwerk wird nun analysiert, indem automatisch folgende Schritte durchgeführt werden.

@ Die Auslastung (utilization) an jeder Station wird berechnet.

- Die Variabilität eines jeden Packet Flows (repräsentiert durch den quadrierten Variationskoeffizienten) beim Verlassen der Stationen entlang seines Routing-Pfades wird bestimmt.
- Es werden Wartezeiten (delays) und Warteschlangenlängen (queue sizes) an den Stationen ermittelt.
- Es werden die globalen Wartezeiten (end-to-end delays) für jeden Packet Flow berechnet.

Die Grundlagen für diese Methode wurden von Kühn, Labetoulle, Pujolle und anderen für komplexe Warteschlangennetze mit Rückkoppelung geschaffen [Kühn79]. Die Variabilität bez. Ankunftsströmen und Paketgrößen ist durch spezielle Gleichungssysteme erfassbar: Die „Austrittsvariabilität" eines jeden Paket Flows an jeder Station entlang seines Routing-Pfades wird beschrieben als lineare Funktion der Ankunftsraten und Variationskoeffizienten aller an einer Station eintreffenden Packet Flows. Für gößere, praxisrelevante Netzwerke sind diese Gleichungen nicht mehr lösbar, das Problem ist NP-hart. Daher wird im Predictor ein Dekompositionsverfahren verwendet, welches die wechselnden Variabilitäten in den Stationen eines jeden Packet Flows approximativ berechnet. Die Variabilität im Netzverkehr hat entscheidenden Einfluss auf Wartezeiten, Warteschlangenlänge sowie insbesondere auf die Antwortzeiten. Die Auslastung bleibt dagegen unberührt. Vergleiche mit Simulationsexperimenten haben gezeigt, dass Dekompositionsverfahren unter folgenden Bedingungen recht genaue Ergebnisse liefern [Kühn79]:

- Geringer oder hoher Verkehr,
- hohe Variabilität der Ankunftsströme und der Bedienzeiten,
- hohe Netzwerkkomplexität und
- geringe „Geschlossenheit" des globalen Netzwerks.

Bei den Predictor-Modellen kommt hinzu, dass wir keine Rückkoppelungseffekte haben, was sich günstig auf Effizienz und Genauigkeit der Lösungsalgorithmen auswirkt. Behält man diese Regeln im Hinterkopf, so lassen sich mit dem Predictor Modelle erstellen, die für Planungszwecke – und mit dieser Intention wird der Predictor ja eingesetzt – hinreichend genaue Entscheidungsunterstützung bieten.

04.06 Weitere Werkzeuge

Wie bereits erwähnt ist der Predictor, der ursprünglich von CACI[16] entwickelt wurde, unter dem neuen Namen EcoPredictor Teil der EcoSYSTEMS Werkzeugfamilie[17]. Ein Bestand-

[16] http://www.caciasl.com/
[17] http://www.compuware.com/products/ecosystems/

teil des Predictors ist auch der Baseliner, welcher den Import von Mess- und Monitoringdaten zur (halb-) automatischen Erstellung von Basismodellen erlaubt. Der COMNET Simulator (COMNET III) erlaubt komplexere und detailliertere Beschreibungen, insbesondere auch Lastbeschreibungen, welche sich nicht nur auf die Paketebene erstrecken.

Weitere Beispiele für kommerziell verfügbare Werkzeuge sind von MIL3[18] die Werkzeuge Modeler, der IT DecisionGuru, der Application DecisionGuru und NETbiz sowie von HyPerformix[19] (ehemals SES) der Strategizer für Client/Server Kapazitätsplanung. Weitere Informationen findet man auf der Homepage des MAPKIT-Projektes [MAP2000].

04.07 Referenzen

[ECO2000] http://www.compuware.com/products/ecosystems/

[HaZo95] Haas, M., Zorn, W.; Methodische Leistungsanalyse von Rechensystemen; Oldenbourg Verlag, München Wien, 1995

[Kühn79] Kuehn, P.; Approximate Analysis of General Queuing Networks by Decomposition; IEEE Transactions on Communications, Vol. Com. 27, No. 1, January 1979

[MakeSys98] http://www.makesystems.com/Pages/WPaper.html#process
Network Resource Planning Process

[MAP2000] Anforderungen und Vorgehensmodell für das Kapzitätsmanagement, MAPKIT-Projektstudie,
http://mapkit.informatik.uni-essen.de/

[Mena94] Menascé, D. A., Almeida, V. A. F., Dowdy, L. W.; Capacity Planning and Performance Modeling – From Mainframes to Client-Server Systems; Prentice Hall, 1994

[Pred97] COMNET Predictor Getting Started Guide; Release 1.3, May 1997

[18] http://www.mil3.com/
[19] http://www.hyperformix.com

KAPITEL 05 DAS SIMULATIONSTOOL COMNET-III

TORSTEN BERNERT, JÖRG HINTELMANN

05.01 Überblick

Das Simulationswerkzeug COMNET III ist in unter Berücksichtigung eines objektorientierten Designs in der Simulationssprache MODSIM II implementiert worden. CACI kann hier auf eine über 30-jährige Erfahrung bei der Entwicklung von Simulatoren zurückgreifen.

Das nächste Kapitel bildet den Hauptteil dieser Evaluationsstudie und erläutert die Modellierungsmöglichkeiten, die sich einem in COMNET III bieten. Um den Rahmen an dieser Stelle nicht zu sprengen, werden ausschließlich die grundlegenden Fähigkeiten des Simulators besprochen. Ein kleines Beispielmodell, ausführlicher vorgestellt in der Evaluationsstudie zum COMNET Predictor, wird erst später, bei der Behandlung der Ergebnisrepräsentation, hinzugezogen.

Kapitel 05.06 bewertet ausgewählte und bezüglich der Modellierung von Netzwerken und Rechensystemen relevante Kriterien in Hinblick auf das hier vorgestellte Werkzeug COMNET III.

05.02 Modellierungsansätze

Sämtliche Komponenten, die in einer realen Systemlandschaft zu finden sind – z.B. Computer, Router, Switches, Netztypen usw. – sind auch im COMNET III-Simulator in Form von Icons präsent, und wie man es von Objekten erwartet, lassen sie sich in ihren Charakteristiken weitestgehend den jeweiligen Erfordernissen anpassen. Eine wesentliches Merkmal, das all die verschiedenen Objekte, welche in den nächsten Abschnitten noch genauer betrachtet werden, auszeichnet, sind statistische Verteilungen. Diese erlangen umso mehr an Bedeutung, wenn man sich das Prinzip des Simulationsablaufs von COMNET III vor Augen führt: Es handelt sich um eine ereignisgetriebene, stochastische, dynamische Simulation. Die initialen Ereignisse und die Zeitpunkte aller Folgeereignisse werden auf der Basis von Zufallszahlen, die aus diesen Verteilungen „gezogen“ werden, ermittelt. Viele weitere Parameter, die in den nächsten Abschnitten zur Sprache kommen, basieren ebenfalls auf Verteilungen und daraus gewonnenen Zufallszahlen, jedoch wird darauf nicht ständig explizit hingewiesen, da dies bei Simulationswerkzeugen ein etabliertes Konzept

ist. Ein multiplikativer Kongruenzgenerator erzeugt in COMNET III Zufallszahlen zwischen 0.0 und 1.0. Mit Hilfe gewichteter Funktionen werden daraus Zufallsvariablen diverser Verteilungen gewonnen. Bis zu insgesamt 99 unabhängige Zufallszahlengeneratoren können von den Verteilungen in einem Modell genutzt werden, indem der zu jeder Verteilung gehörende Parameter *Stream* auf einen noch frei belegbaren Stream gesetzt wird; initial benutzen alle Verteilungen den Stream 0 – natürlich mit unterschiedlichen Seeds.

Die folgenden Abschnitte erläutern die wesentlichen Schritte der Erstellung eines COMNET III-Modells. In Abschnitt 05.03 wird das Topologiemodell behandelt, in Abschnitt 05.04 das Lastmodell, Abschnitt 05.05 zeigt die Möglichkeiten der Ergebnisauswertung und -präsentation.

05.03 Topologiemodell

Der erste Schritt der Modellierung besteht in der Erstellung eines Topologiemodells, welches ein existierendes oder geplantes System[20] in geeigneter Weise, d.h., so abstrakt wie möglich und so detailliert wie nötig, abbilden soll. Die wesentlichen Gestaltungmittel für diese Aufgabe stehen in Form von „Nodes“ und „Links“ bereit, die aus einer Werkzeugpalette auf die Arbeitsfläche gezogen werden. Diese repräsentieren die bereits erwähnten Computer, Router, Switches usw. im Falle der Nodes, bzw. diverse Netztypen mit CSMA/CD-, Token-passing-Verfahren usw. im Falle der Links. Alternativ zum „Modellieren auf der grünen Wiese“ können Topologiedaten, die mit Netzwerk-Management-Systemen erfaßt wurden, in den Simulator importiert werden. Ein Import von Predictor-Modellen ist ebenfalls möglich.

(a) Nodes

In COMNET III-Modellen können drei verschiedene Node-Typen Verwendung finden: Processing Nodes, Network Devices und Computer Groups. Bei den beiden erstgenannten Typen können charakteristische Eigenschaften mittels Zufallszahlen beschrieben werden. Beispielhaft seien *Time to failure* oder *Time to repair* genannt. Treffen, wie noch in Abschnitt 05.04 besprochen wird, Lasten auf einen Node (z.B. Applikationen), so werden die in diesen Lasten geforderten *Commands* im Node abgearbeitet, sofern dieser frei ist; ansonsten werden sie vorerst in der Warteschlange des Nodes gepuffert. Solche Commands sind beispielsweise: *Process Data*, *Answer Message*, *Read* und *Write File* usw. Jedes Command wird im Node anders behandelt und verbraucht hier per Zufallsvariable(n) definierte Zeit(en).

[20] Der Begriff „System“ wird in diesem Papier oft als Synonym für „Netzwerk“ im weitesten Sinne verwendet. Sämtliche Hard- und Softwarekomponenten, die ein Netzwerk ausmachen, sind in diesen Begriff eingeschlossen.

Die drei Node-Typen und ihre grundsätzlichen Eigenschaften sollen noch kurz dargelegt werden:

- Ein *Processing Node* repräsentiert einen beliebigen Computer (Client, Server, Mainframe usw.). Wesentliche Parameter, die diesen Note-Typ näher beschreiben sind: In- und Out-Port-Delays inklusive vorgesehener Puffergrößen, Anzahl der Prozessoren und weitere Parameter wie z.B. *Time Slice* oder *Selection Rules* – also Bedienstrategien sowie typische Festplattencharakteristika.
- Ein *Network Device* ist für alle Arten von Switches, Bridges, Routers, Hubs und vergleichbare Netzwerkkomponenten zuständig. Es existiert eine umfangreiche Bibliothek mit vorparametrisierten Geräten bekannter Hersteller. Zusätzlich zu den beim Processing Node vorgestellten Parametern kann für ein Network Device die interne *Bus Rate* in Mbps (Megabits per second) und die *Bus Count* angegeben werden, welche im Wesentlichen den Durchsatz des Gerätes zwischen den Ports bestimmt.
- Eine *Computer Group* repräsentiert eine beliebige Anzahl identischer Computer und kann so helfen, den Modellierungsaufwand erheblich zu reduzieren.

Subnets sind ein Mittel zur Hierarchisierung großer Netzwerkstrukturen. Sie repräsentieren ganze Teilnetze auf der darüberliegenden Modellebene als *einen* Node.

(b) Links

Links werden benutzt, um verschiedenartige Übertragungsmedien zu modellieren. Die physikalischen Charakteristika eines Links werden durch *Bandwidth* und *Propagation Delay* beschrieben. Auf Anwendungsebene definierte Pakete werden in Links (gemäß Verbindungsschicht des ISO/OSI-Protokolls) gegebenenfalls segmentiert oder zusammengesetzt. Jeder Link hat eine *Frame Size (minimum* und *maximum)*, einen *Frame Overhead* und eine *Frame Error Rate*.

In der COMNET III-Bibliothek sind diverse vorparametrisierte Link-Typen zu finden. Die Parameter beruhen auf den entsprechenden Standards:

- *CSMA/CD*-basierte Netze nach IEEE 802.3(z)
- *Token-passing*-Verfahren nach IEEE 802.4/802.5
- *Point-to-point*-Links für Verbindungen wie ISDN, SONET usw.
- *WAN Link or VC (Virtual Circuit)* für WAN-Dienste auf einer höheren Abstraktionsebene. Dies bietet sich an, wenn Teile des Netzwerks (Topologie und/oder Lasten) unbekannt oder wenig bedeutsam für das Modell sind.

Transit Networks sind ein Mittel zur Hierarchisierung großer Netzwerkstrukturen. Sie repräsentieren ganze Teilnetze auf der darüberliegenden Modellebene als *einen* Link.

05.04 Lastmodell

Eine einfache Art, Datenflüsse zu modellieren, sind sogenannte *Packet Flows*. Dieses Konzept wird in der Evaluationsstudie des COMNET Predictor vorgestellt und soll aus diesem Grunde hier nicht redundant behandelt werden. Die Möglichkeiten in COMNET III sind zwar leicht erweitert, grundsätzlich jedoch vergleichbar. *Ein* Packet Flow kann nur durch *einen Quellknoten* und *einen Zielknoten* beschrieben werden.

Datenflüsse werden in COMNET III üblicherweise durch Verkehrsquellen, *Traffic Sources*, modelliert und mit den entsprechenden Nodes verbunden. Ein Node kann auf diese Weise mehrere verschiedenartige Lasten oder Aufrufe, *Calls*, erzeugen, jedoch kann eine Verkehrsquelle nur mit jeweils einem Node assoziiert werden. Fünf unterscheidbare Verkehrsquellen stehen für die Modellierung von Datenflüssen zur Verfügung: Message Source, Response Source, Application Source, Session Source und Call Source. *Eine* Source kann im Gegensatz zu einem Packet Flow *mehere Zielknoten* auf einmal ansprechen.

@ Mit Hilfe der *Message Source* ist es möglich, Nachrichten an ein oder mehrere Ziele zu versenden. Viele einfachere Formen des Datentransports, wie z.B. File Transfer oder eMail-Verkehr, lassen sich auf diese Weise modellieren. Im Eigenschaften-Dialog kann die Charakterisik einer Nachchrichtenquelle näher beschrieben werden.

@ Die *Response Source* ist ein Nachrichtengenerator, der nach empfangener Nachricht aktiv wird. Dies ist beispielsweise nützlich bei Datenbankanfragen oder eMail-Replies. Die generierte Antwort wird stets an den Node gesendet, von dem aus die jeweilige Response Source aktiviert wurde.

@ *Application Sources* werden in COMNET III genutzt, um aufwändigere Lasten in Nodes zu erzeugen, Lasten, die sich aufgrund komplexer Prozesse in einem Node ergeben. Die Prozesse werden Skript-artig unter Verwendung der vorne bereits erwähnten *Commands* beschrieben. Skripts können einfache Kontrollflusskonstrukte wie *if*, *while*, *repeat* verwenden. Wenn eine Application Source auf einem Node zur Ausführung gelangt, so werden sämtliche Commands des Skripts sequentiell abgearbeitet, bis die Applikation, oder besser: das abstrakte Abbild der Applikation, beendet ist.

@ Die *Session Source* ist ein Nachrichtengenerator, der zuerst eine Session-Verbindung zu einem anderen Node etabliert. Innerhalb einer Session lassen sich nun verschiedenartige Nachrichten versenden – der Overhead durch einen ansonsten ständig zu wiederholenden Verbindungsaufbau entfällt. Diese Quelle ist damit in erster Linie für die Modellierung eines verbindungsorientierten Verkehrs geeignet.

@ *Call Sources* generieren circuit-switched oder packet-switched calls. Eine Call Source spezifiziert diese Aufrufe in Form von Zwischenankunftszeiten, Dauer und Bandbreitenanforderungen. Wann immer eine Call Source während des Simulationslaufs aktiviert wird, wird versucht, eine dedizierte End-to-end-Verbindung zwischen den betrof-

fenen Nodes zu etablieren unter Berücksichtigung der geforderten Bandbreitenanforderungen entlang des gesamten Routing-Pfades. Erst wenn die Bandbreitenanforderungen an jedem Node und Link des Pfades zugesichert wurden, kann der Pfad für die vereinbarte Dauer benutzt werden. Für alle übrigen Datenflüsse stehen für diese Dauer nur die entsprechend reduzierten Bandbreiten an den betroffenen Komponenten zur Verfügung.

Die Art und Weise der Übertragung von derart generierten Daten regeln in realen Netzwerken wie in den Modellen des COMNET III Protokolle. Die beiden wichtigsten, das Transportprotokoll und das Routingprotokoll, sollen kurz vorgestellt werden:

@ End-to-end- oder Transport-layer-Funktionen des OSI-Modells werden durch das *Transportprotokoll* realisiert, das mit jeder versendeten Nachricht assoziiert ist. Wichtige Transportprotokolle wie ATP, TCT/IP, NetBios, UDP/IP usw. sind in der erweiterbaren Bibliothek vorparametrisiert. Wesentliche Parameter dieses Protokolls sind: Paketgröße, Overhead, Kontrollflussmechanismus oder die Fähigkeit, blockierte Pakete erneut zu übertragen. Anhand dieser Werte lassen sich Aufwände für die Segmentierung und das Wiederzusammensetzen von Paketen in Leistungsmaßen wie beispielsweise End-to-end-Delays (Antwortzeiten) erfassen.

@ Netzwerk-layer-Funktionen werden in Nodes realisiert, in denen die Routing-Entscheidungen getroffen werden. Für jedes Paket kann ein Routing-Verfahren bestimmt werden. Auf diese Weise lassen sich Bedienzeiten an Ports und in den Nodes beeinflussen. Die Routing-Entscheidungen in jedem Node basieren auf Routing-Algorithmen. Es existieren:

- *Statische Algorithmen.* Sie berechnen die Routing-Tabelle einmalig zu Beginn der Simulation.
- *Dynamische Algorithmen*. Sie berechnen die Routing-Tabelle periodisch in Abhängigkeit von dynamischen Maßen.
- *Eigene Routing-Tabellen*. Hier lassen sich eigene Einträge in die Routing-Tabelle vornehmen und die Selektionskriterien für die Routing-Wahl spezifizieren.

Die beiden erstgenannten belegen die Einträge der Routing-Tabelle automatisch anhand von Algorithmen, die den kürzesten Routing-Pfad berechnen. Multipath Routing, das einen Lastausgleich auf alternativen Pfaden unter Berücksichtigung verschiedener Metriken erreicht, läßt sich ebenfalls modellieren. Beide Verfahren aktualisieren die Routing-Tabellen, wenn ein Node oder Link ausfällt oder seine Dienste nach verstrichener Reparaturzeit wieder anbietet.

Weiterhin existieren Commands, die Ereignisse auf Session- oder Präsentations-/Applikations-Ebene modellieren können.

05.05 Ergebnispräsentation

Ist das Modell fertiggestellt, kann es simuliert werden. Um dies kurz zu veranschaulichen, wird das Beispielmodell der Predictor-Evaluationsstudie in den COMNET III-Simulator importiert. Zur Erinnerung werden die Topologie ((a) – Oberstes Level, (b) – Subnet „Intranet“) und die Packet Flows noch einmal präsentiert.

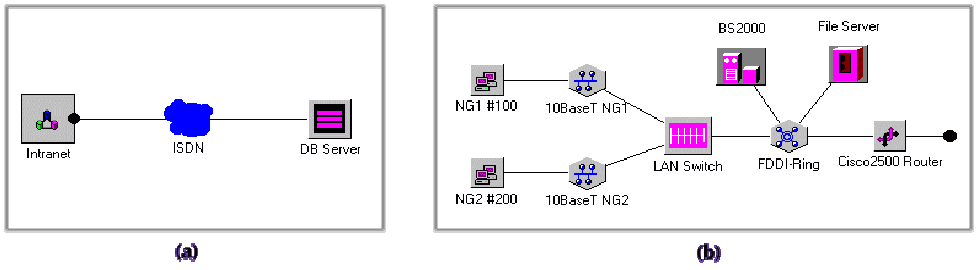


Abbildung 05-1: Topologien des Modells

Scenario	Origin	Destination	Application	Protocol	Packets/Sec	Burstiness	Avg Bytes/Pkt	StDev Pkt Size	Bytes/Sec
Baseline	DB Server	Intranet.NG1 #100	DB-Transaktion	IP	0,16666	1,44	5000	0	833,3
Baseline	DB Server	Intranet.NG2 #200	DB-Transaktion	IP	0,16666	1,44	20000	0	3333,2
Baseline	Intranet.BS2000	Intranet.NG2 #200	Host-Appl	IP	50	6,25	6000	2000	300000
Baseline	Intranet.File Server	Intranet.NG1 #100	File Transfer	IP	10	1	50000	0	500000
Baseline	Intranet.NG1 #100	DB Server	DB-Transaktion	IP	0,16666	1,44	400	0	66,664
Baseline	Intranet.NG1 #100	Intranet.File Server	File Transfer	IP	5	1	1000	0	5000
Baseline	Intranet.NG2 #200	DB Server	DB-Transaktion	IP	0,16666	1,44	400	0	66,664
Baseline	Intranet.NG2 #200	Intranet.BS2000	Host-Appl	IP	50	6,25	500	100	25000

Abbildung 05-2: Definition von Paketflüssen

Die Topologie wird vom Simulator 1:1 übernommen, einige Parameter in den Packet Flows leicht abgewandelt. Beispielsweise werden die allgemeinen Verteilungen der Ankunftsrate mit einer Burstiness ungleich Eins in entsprechend parametrisierte Gamma-Verteilungen umgesetzt, Paketgrößen mit einer Standardabweichung größer Null in äquivalente Normalverteilungen.

Als nächstes gilt es die Objekte auszuwählen, über die Statistiken geführt werden sollen ...

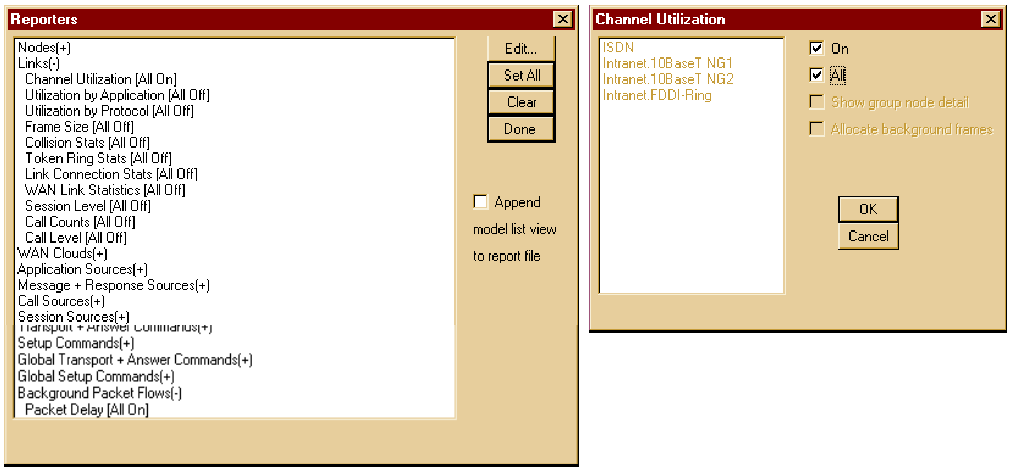


Abbildung 05-3: Auswahl von Objekten zur statistischen Auswertung

und den Simulationslauf zu konfigurieren ...

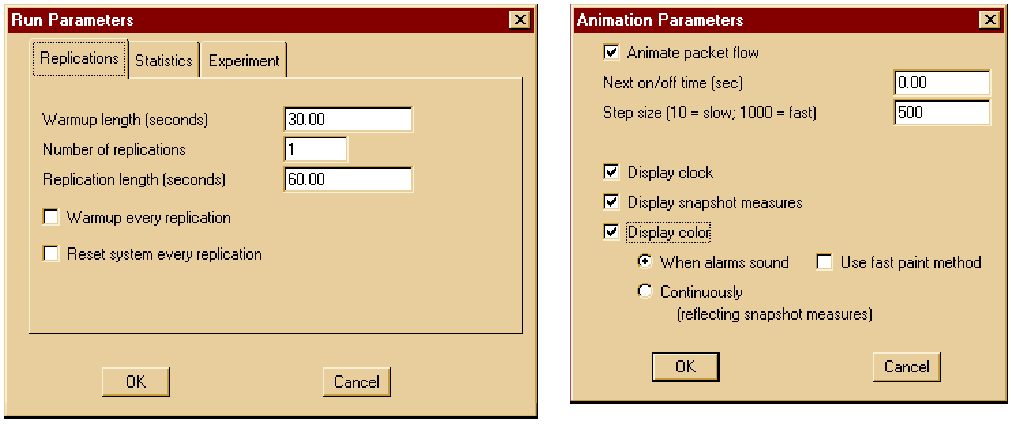


Abbildung 05-4: Konfigurierung eines Simulationslaufs

Dann kann die Simulation gestartet werden. Mit den oben zu sehenden Einstellungen: Transiente Phase = 30 s, *ein* Simulationslauf von 60 s und eingeschalteter Animation rechnet ein AMD K6-2 3D, 300 MHz 3:50 min. Die gesammelten Statistiken werden als Report in eine ASCII-Datei geschrieben; sie hat nach dem Simulationslauf des Beispielmodells folgenden Inhalt.

CACI COMNET III Release 2.0 Build 722k (Academic license) Tue Jul 27 12:51:38 1999
PAGE   1

BspNetz

NODES: NODE FULL UTILIZATION

REPLICATION 1 FROM 30.0 TO 90.0 SECONDS

NODES: / PORTS:	BUSY CENTRAL PROCESSORS BUSY INPUT PROCESSORS MEAN	MAX	UTIL%	BUSY BUSES BUSY OUTPUT PROCESSORS MEAN	MAX	UTIL%
DB Server	.0015667	1	.1566667	N/A	N/A	N/A
ISDN	0.0	0	0.0	0.0	0	0.0
Intranet.BS2000	.1991333	1	19.9133	N/A	N/A	N/A
Intranet.FDDI-Ring	0.0	0	0.0	0.0	0	0.0
Intranet.File Server	.0315000	1	3.1500	N/A	N/A	N/A
Intranet.FDDI-Ring	0.0	0	0.0	0.0	0	0.0
Intranet.LAN Switch	0.0	0	0.0	.0083248	1	.8324756
Intranet.10BaseT NG1	0.0	0	0.0	0.0	0	0.0
Intranet.10BaseT NG2	0.0	0	0.0	0.0	0	0.0
Intranet.FDDI-Ring	0.0	0	0.0	0.0	0	0.0
Intranet.Cisco2500 Rou	.0001000	1	.0100000	.0000866	1	.0086560
Intranet.FDDI-Ring	0.0	0	0.0	0.0	0	0.0
ISDN	0.0	0	0.0	0.0	0	0.0

CACI COMNET III Release 2.0 Build 722k (Academic license) Tue Jul 27 12:51:38 1999
PAGE   2

BspNetz

LINKS: CHANNEL UTILIZATION

REPLICATION 1 FROM 30.0 TO 90.0 SECONDS

LINK	FRAMES DELIVERED	RST/ERR	TRANSMISSION DELAY (MS) AVERAGE	STD DEV	MAXIMUM	% UTIL
ISDN						
FROM Intranet.Cisco2	24	0	50.000	0.000	50.000	2.0000
FROM DB Server	24	0	1640.625	934.239	2500.000	65.04
Intranet.10BaseT NG1	21225	0	1.638	20.905	1302.951	42.23
Intranet.10BaseT NG2	16183	0	2.066	22.016	1222.293	25.85
Intranet.FDDI-Ring	16221	0	0.249	0.141	0.360	6.7290

```
CACI COMNET III  Release 2.0 Build 722k (Academic license)  Tue Jul 27 12:51:38 1999
PAGE     3

                                  BspNetz

                    BACKGROUND PACKET FLOWS: PACKET DELAY

                    REPLICATION 1 FROM 30.0 TO 90.0 SECONDS

ORIGIN, APP, PROTOCOL             NUMBER OF PACKETS           PACKET DELAY (MS)
  DESTINATION            CREATED DELIVERED RESENT DROPPED   AVERAGE    MAXIMUM
_____________________    _______ _________ ______ _______  _________  _________


Intranet.NG1 #100, DB-Transaktion, IP
  DB Server                   15        15      0       0    140.166    1265.733

Intranet.NG2 #200, DB-Transaktion, IP
  DB Server                    9         9      0       0     67.039     182.508

Intranet.NG2 #200, Host-Appl, IP
  Intranet.BS2000           3135      3135      0       0     10.926    1224.344

Intranet.NG1 #100, File Transfer, IP
  Intranet.File Server       332       332      0       0     26.941    1305.088

DB Server, DB-Transaktion, IP
  Intranet.NG1 #100           11        11      0       0   1996.939    6456.777

Intranet.File Server, File Transfer, IP
  Intranet.NG1 #100          613       612      0       0     62.624     181.905

DB Server, DB-Transaktion, IP
  Intranet.NG2 #200           12        13      0       0   4526.372    9044.112

Intranet.BS2000, Host-Appl, IP
  Intranet.NG2 #200         2839      2836      0       0     19.675     117.553

**********************************************************************
* This report was generated by an academic license of COMNET III, *
* which is to be used only for the purpose of instructing        *
* students in an accredited program that offers AA, bachelors, or *
* graduate degrees.  The information in this report is not for    *
* commercial use, funded projects, funded research, or use for    *
* the benefit of any external organization.                       *
**********************************************************************
```

Tabelle 05-1: Tabellarische Ausgaben

Die grafische Repräsentation der Ergebnisse ist in COMNET III nicht vorgesehen. Hier kann und muss man sich mit dem Export der Simulationsstatistiken im xpt-Format (vergleichbar dem csv-Format, Werte jedoch durch Tabs getrennt) weiterhelfen.

05.06 Ausgewählte Kriterien

Das abschließende Kapitel erläutert stichpunktartig ausgewählte Kriterien. Es existieren auch Faktoren, die die Relevanz der einzelnen Kriterien unterschiedlich stark gewichten, diese werden jedoch an dieser Stelle nicht aufgeführt.

1. ***Kosten:***
   Eine kommerzielle COMNET III-Lizenz kostet zur Zeit (August 1999) pro Arbeitsplatz ca. *35.000 $*. Universitäts-Editionen sind, bei eingeschränkter Funktionalität (max. 20 Nodes dürfen Lasten generieren oder empfangen), entschieden günstiger – Verhandlungsgeschick ist gefragt. Kostenlose Demo-Versionen sind erhältlich, jedoch nicht auf anonymem Wege. Es muss ein Key für die *Trial License* angefordert werden.

   Der COMNET III-Simulator läuft auf folgenden Plattformen: *Windows NT/9x*, *Solaris 2.5* und höher, *HP-UX*. Zusammen mit der COMNET III-Lizenz wird ein *Wartungsvertrag* erworben. *Schulungsangebote* werden ständig auf den Internetseiten von CACI angeboten.

2. ***Modellierungsparadigma***
   Im Predictor werden Netzwerkmodelle *grafisch* unter Verwendung von Bausteinen erstellt, die *hierarchische Modellierung* wird unterstützt. Lasten werden im Allgemeinen durch Source-Objekte modelliert, die je nach Art der Lasterzeugung verschiedenartig parametrisiert werden können. In einfacher Form können Lasten auch in eine Tabelle eingetragen werden.

3. ***Lösungstechniken***
   COMNET III simuliert ein Netzwerkmodell auf der Basis einer ereignisorientierten, stochastischen, dynamischen, diskreten *Simulation*. Die Parametrisierung der Objekte erfolgt jedoch eher auf Prozess- anstatt auf Ereignisebene. Ein inhärentes Problem der Simulation, von der auch COMNET III nicht verschont bleibt, ist ihre *Effizienz* und die *Laufzeitanforderung* bei der detaillierten Berechnung großer Netzwerke und vieler Datenflüsse.
   Die *Güte der Lösungen* hängt sehr von den Modelleigenschaften ab. Detaillierte Modelle liefern tendenziell genauere Ergebnisse, haben jedoch negative Auswirkungen auf die Laufzeit. Grobere Modelle weisen naturgemäß größere Fehler auf, so dass auf dieser (abstrakteren) Ebene i.d.R. auf analytische Methoden zurückgegriffen werden sollte.

4. ***Benutzerschnittstelle***
   Die *Bibliotheken* des COMNET III stellen viele vorparametrisierte Bausteine bereit. Insbesondere HW-Komponenten wie Netztypen, Switches, Router u.ä. sind hier zu finden.

Die *Ergebnisdarstellung* ist auf einen textuellen Report beschränkt, der in eine ASCII-Datei geschrieben wird. *Experimentserien* werden durch einen *Automatic Parameter Iteration*-Ansatz unterstützt. Das Modell wird mit entsprechend geänderten Parametern neu gerechnet; je Iteration verdoppelt sich jedoch die Dauer der Berechnung. Die *Anwendbarkeit* des Programms ist nach angemessener Einarbeitungszeit intuitiv.

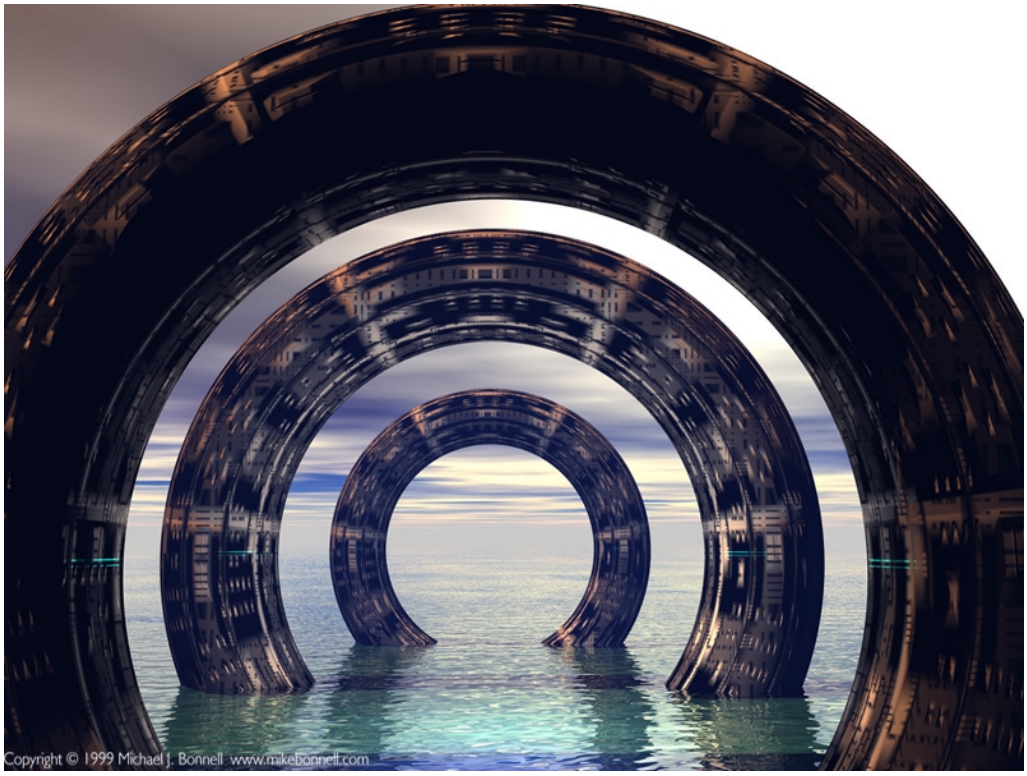
5. ***Integration mit anderen Werkzeugen***
   Mit Hilfe des zur Distribution gehörenden *COMNET Baseliners* lassen sich Tolologie- und Lastdaten, die mit führenden kommerziellen Netzwerk-Management- bzw. Netzwerk-Monitoring-Werkzeugen gewonnen wurden, in den Simulator importieren. Dies kann, insbesondere bei großen Modellen, eine erhebliche Reduzierung des Modellierungsaufwands bedeuten, aber auch die Gefahr in sich bergen, ein Modell mit viel „Ballast" zu erhalten. Mit dem *COMNET Predictor* erstellte Modelle können ebenfalls importiert werden. Export von Simulationsstatistiken und Plot-Daten im *xpt-Format* (vergleichbar dem csv-Format, Werte jedoch durch Tabs getrennt) ist möglich.

6. ***Erweiterbarkeit***
   Auf das Vorhandensein vordefinierter *Bausteinbibliotheken* wurde bereits unter 4. hingewiesen. Diese Bibliotheken sind vom Benutzer erweiterbar. Die *Simulationsalgorithmen* des COMNET III sind „fest verdrahtet".

7. ***Validierungshilfen, Konsistenzprüfung***
   Validierungshilfen bietet das Programm nicht. *Konsistenzprüfungen* überprüfen das Modell-Layout, die Zwischenverbindungen (isolierte Objekte sind nicht erlaubt) und Objektparameter.

8. ***Bemerkungen bez. spezieller Eigenschaften***
   - Die Online-Animation der Datenflüsse ist möglich.
   - Viele Parameter müssen angegeben werden ⇒ Problem der Parametergewinnung.
   - Durch die Vielzahl der Parameter wird eine Genauigkeit des Modells vorgespielt, die nicht immer gegeben sein muss, da sie letztendlich von der Qualität der Parameter abhängt. Dies ist jedoch ein generelles Problem bei der Modellierung.
   - Werkzeuge ermitteln aus empirisch erhobenen Daten automatisch die am besten geeignete Wahrscheinlichkeitsverteilung.
   - Werkzeuge können die Ergebnisse analysieren: berechnen von Konfidenzintervallen, Regressionsanalyse, Analyse der Varianzen mehrerer Replikationsläufe. (Konnte nicht nachvollzogen werden, evtl. wegen beschränkter Universitäts-Lizenz.)

Teil IV: Management Frameworks



Inhalt von Teil IV

KAPITEL 01 CA UNICENTER 3

01.01 PERFORMANCE MANAGEMENT 3
01.02 RESPONSE MANAGER OPTION 6

KAPITEL 02 HP TOOLS 11

02.01 EINLEITUNG 11
02.02 HP MEASUREWARE/PERFVIEW 11
02.03 HP ETHERNET LANPROBE / HP NETMETRIX 14

KAPITEL 03 GLANCEPLUS 20

03.01 BENUTZERSCHNITTSTELLE 20
03.02 HIERARCHISCHE DARSTELLUNG (TOP-DOWN ANALYSE) 20
03.03 DEFINITION VON APPLIKATIONEN 21
03.04 FUNKTIONEN VON GLANCEPLUS 21
03.05 ABGRENZUNG ZU DEN STANDARD-MESSTOOLS IN UNIX 22
03.06 ZUSAMMENFASSUNG 23

KAPITEL 04 BEST/1 – EINE ÜBERSICHT 24

04.01 BETRIEB 24
04.02 DIE KOMPONENTEN VON BEST/1 24
04.03 AUFBAU EINES BEST/1-MODELLS 29
04.04 ZUSAMMENFASSUNG 30

KAPITEL 05 BEST/1 – EIN ERFAHRUNGSBERICHT 32

05.01 EINLEITUNG 32
05.02 DIE IT-UMGEBUNG 32
05.03 DIE SYSTEMS MANAGEMENT KNOTEN 33
05.04 ANFORDERUNGEN DES KUNDEN 33
05.05 DIE REALISIERUNG 35
05.06 KONFIGURATION 36
05.07 DIE ANFORDERUNGEN IN STICHWORTEN 37

KAPITEL 01 CA UNICENTER

HEINER RISTHAUS

01.01 Performance Management

(a) Architektur

Die Performance Managementkomponente von Unicenter TNG benutzt eine typische Manager-Agent-Architektur. Die Managementkomponenten bestehen aus den Teilen

- Performance Configuration
- Performance Scope
- Performance Trend
- Chargeback

und sind ausschließlich für NT verfügbar. Die Performanceagenten umfassen einen historischen und einen Realtime-Agenten und existieren für Windows-, Unix- und Novell-Umgebungen.

Die Konfiguration des Performancemanagements erfolgt durch ein eigenes Werkzeug (Performance Configuration). Zunächst zeigt es alle Computer im Repository, wahlweise können dann allerdings auch nur die angezeigt werden, auf denen ein Performance Agent installiert ist. Die Computer können der besseren Übersicht halber in Gruppen organisiert werden (z.B. nach organisatorischen, geografischen, funktionalen, ... Kriterien).

Ein weiteres Hilfsmittel für die effiziente Konfiguration ist die Definition von Profilen. Darin wird beschrieben und maschinenunabhängig festgelegt, welche Performancedaten erfasst werden sollen, in welchen Intervallen, wie lange die Daten aufgehoben werden sollen (historische Daten), ... Die Profile können dann Computern bzw. Computergruppen zugewiesen werden.

Zusätzlich kann für die verschiedenen Computer festgelegt werden, welche Managementstation für sie bzgl. der Konfiguration zuständig ist, d.h. Performanceprofile an sie verteilen darf. Auf Basis der zugewiesenen Profile werden die entsprechenden Parameter der Computer erfasst und in so genannten Performance Cubes abgelegt. Dieses dreidimensionale Datenmodell beschreibt

- Datentyp
- Aufzeichnungszeitraum innerhalb eines Tages
- verschiedene Tage bzw. verschiedene Maschinen

Es gibt verschiedene Arten von Cubes:

- tägliche (fein granuliert)
- periodische (über längere Zeiträume)
- Enterprise (über mehr als eine Maschine)

Daher sind die Performancecubes hauptsächlich für die Erzeugung von Langzeitstatistiken und die Betrachtung historischer Daten geeignet. Die Visualisierung der Daten kann durch Performance Scope erfolgen. Damit sind Darstellungen der historischen Verläufe und ebenfalls Echtzeitbeobachtungen aller konfigurierten Parameter möglich. Die Darstellung erfolgt in einem geteilten Graphen, der auf der einen Seite Echtzeit-, auf der anderen Seite statistische Daten für einen vergangenen Zeitraum zeigt.

Ebenso können die zu überwachenden Parameter mit Schwellwerten belegt werden, bei deren Überschreitung Alarme ausgelöst und Aktionen (z.B. Recoverymaßnahmen) gestartet werden können Das Aussehen der Grafiken kann in weiten Bereichen den Erfordernissen oder dem persönlichen Geschmack angepasst werden (Farben, zwei- bzw. dreidimensional, Betrachtungwinkel, Zeitraum, Pollingintervalle bei Echtzeitbetrachtung, ...).

Auswahl und Anzeigeformat von Performanceparametern können in Templates gespeichert werden, so dass sie als Basis für die Erzeugung weiterer Graphen (z.B. für unterschiedliche Rechnersysteme) benutzt werden können.

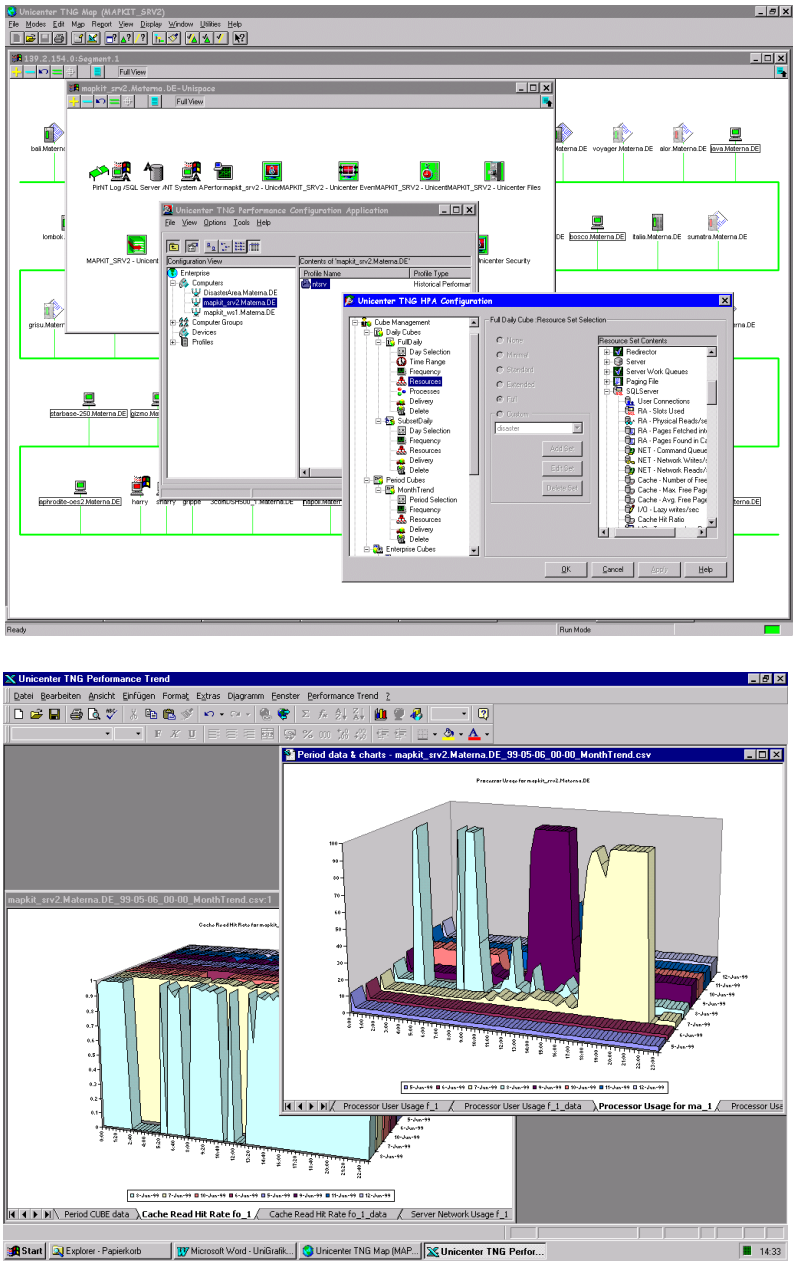


Abbildung 01-1: Performance Trend

Die Performancedaten können für die Erkennung von Trends nach Excel exportiert und dort weiter ausgewertet werden (dazu stellt Performance Trend entsprechende Funktionen zur Verfügung). Ebenso können sie als Grundlage für ein verbrauchsorientiertes Accounting dienen (Chargeback). Auch dazu stehen Excelmakros zur Verfügung, die die Daten entsprechend aufbereiten und Berichte bzw. Abrechnungen generieren. D.h. für die Funktionen Performance Trend und Chargeback ist die Installation eines Microsoft Excel-Programms notwendig.

01.02 Response Manager Option

Die Response Manager Option besteht aus zwei Komponenten: RMON Analysis und Service Analysis. Diese beiden Applikationen setzen auf zwei verschiedenen Verfahren auf.

Zum einen kann die RMON-Analyse mit Hilfe von RMON-fähigen SNMP-Probes durchgeführt werden. Dazu sind spezielle Komponenten innerhalb von Hubs, Routern, ... oder auch dedizierte Stand-Alone-Probes (z.B. von HP) notwendig. Diese sind in der Lage, autonom den Datenverkehr auf ausgewählten Netzsegmenten aufzuzeichen, zu filtern und bei Bedarf Alarme zu generieren. Bei Bedarf werden diese Daten durch die Managementstation abgerufen und zur Auswertung übertragen. Dies geschieht vorzugsweise zu Zeiten, an denen das Netz durch wenig Verkehr belastet ist. Zum anderen können auf Standard-PCs so genannte Response Manager Probes (Software) installiert werden. Diese Probes versetzen den Netzadapter des Computers in den so genannten "promiscious mode". Er wird so dazu veranlasst, alle Pakete auf dem jeweiligen Netzsegment zu betrachten. Die aufgenommenen Pakete werden analysiert und bestimmten Services oder Applikationen zugeordnet.

Eine Reihe von Applikationen und Protokollen sind standardmäßig definiert (z.B. telnet, http, Novell NetWare, Banyan-Vines, ...). Da eine Applikation in diesem Zusammenhang im allgemeinen als eine Kombination aus IP-Adresse und Portnummer verstanden wird, können so neben den vordefinierten Applikationen recht einfach neue definiert werden (sofern die Portnummer bekannt und sinnvollerweise auch fest eingestellt ist).

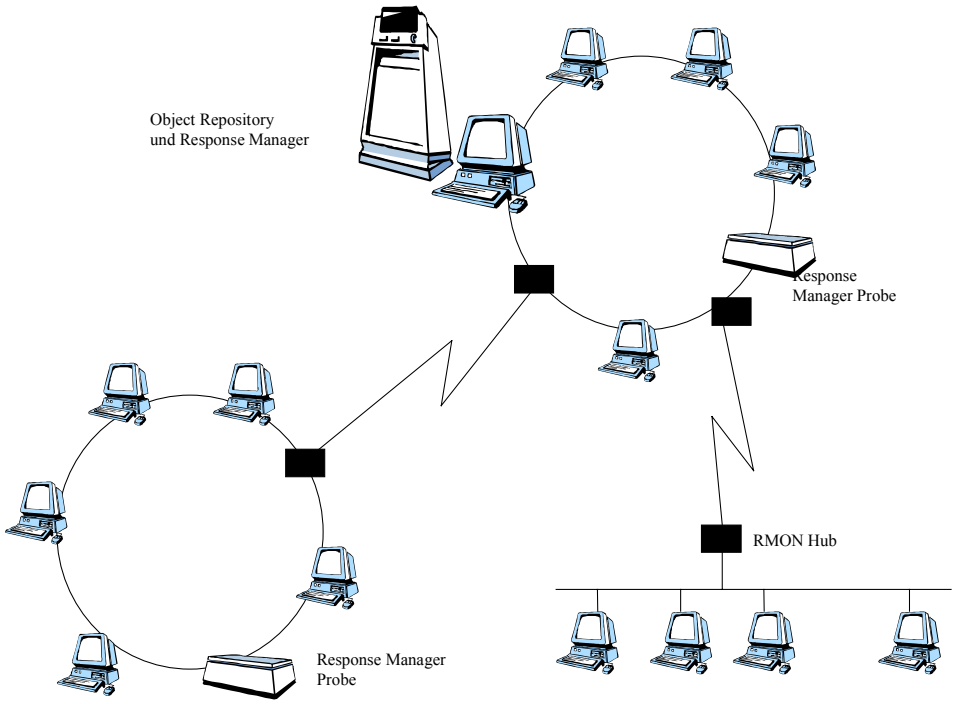


Abbildung 01-2: Response Management Architektur

Die Probe analysiert nun den Datenverkehr auf dem Netzsegment und speichert die gefundenen Daten. Anhand der erkannten Namen, Adressen und Protokolle bzw. Applikationen bestimmt die Probe, welche Systeme und welche Services auf dem Netz verfügbar sind, z.B. LAN Manager, HTTP, ...

Im Allgemeinen werden die Daten zwar regelmäßig, aber in relativ großen Zeitabständen erfasst. Darüber werden dann statistische Auswertungen durchgeführt. Werden bestimmte Schwellwerte dabei überschritten, geht die Probe für einen Zeitraum von zwei Minuten in einem Detailmodus, der für die betroffenen Komponenten Daten mit einer hohen Detailtiefe erfasst. So wird zum einen ein Alarm generiert, der auf die kritische Situation aufmerksam macht, zum anderen wird die nachträgliche Analyse der Situation ermöglicht.

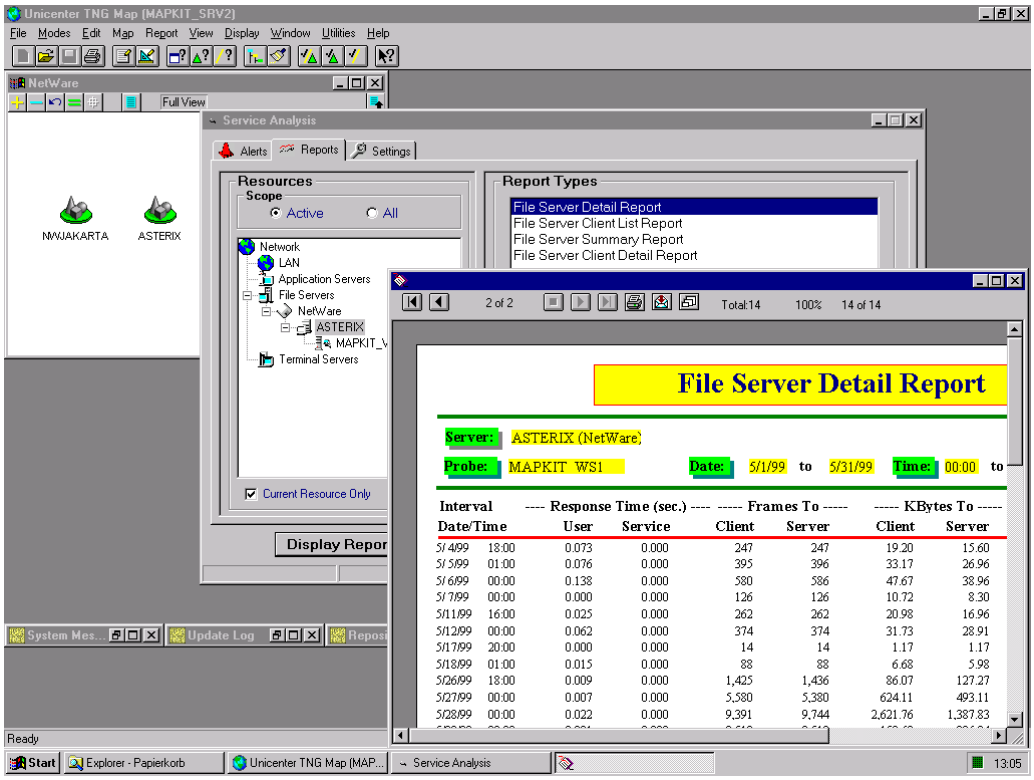


Abbildung 01-3: Service Analysis

Die Schwellwerte werden initial durch ein automatisches Baselining der Probe ermittelt und dynamisch angepasst, können aber auch individuell konfiguriert werden, um sie besonderen Gegebenheiten anzupassen. Im Testnetz hat sich herausgestellt, dass die automatisch generierten Werte brauchbare Informationen liefern, d.h. einen geeigneten Kompromiss aus Dichte der Problemmeldungen einerseits und dem gesamten Informationsgehalt andererseits darstellen.

Die Analysekomponenten errechnen automatisch Statistiken verschiedener Parameter wie z.B. Antwortzeiten, Fehlerraten oder übertragene Datenmengen. Dazu werden am Ende eines Auswerteintervalls die Anzahl der vorgenommenen Messungen, Summe und Quadratsumme aller Messwerte und die maximale Antwortzeit übermittelt, um Mittelwerte und Abweichungen bestimmen zu können.

Die erfassten und berechneten Informationen können dann übersichtlich grafisch und numerisch dargestellt und in verschiedenen Dateiformaten exportiert werden. Damit können

sie z.B. durch zusätzliche Werkzeuge von Drittherstellern visualisiert oder analysiert werden. Zusätzlich können die Informationen z.B. im Inter-/Intranet für Marketingzwecke veröffentlicht werden.

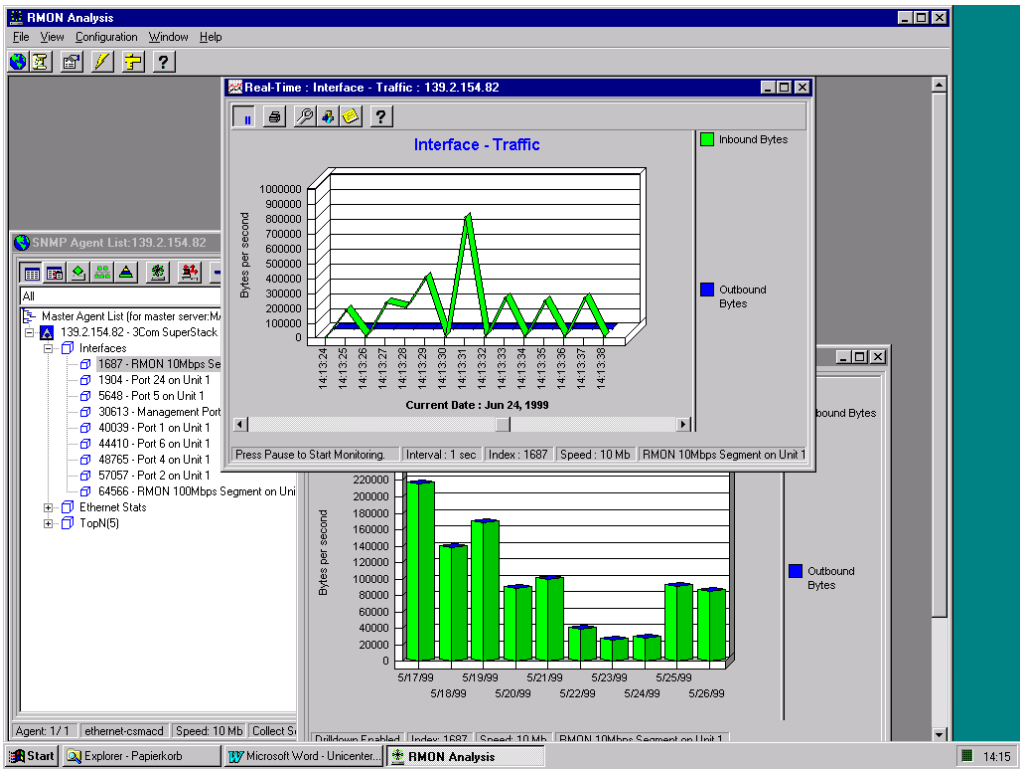

RMON Analysis
File View Configuration Window Help
Real-Time : Interface - Traffic : 139.2.154.82
Interface - Traffic
Inbound Bytes
Outbound Bytes
Bytes per second
Current Date : Jun 24, 1999
Press Pause to Start Monitoring.
Interval : 1 sec
Index : 1687
Speed : 10 Mb
RMON 10Mbps Segment on Unit 1
SNMP Agent List:139.2.154.82
Master Agent List (for master server:M
139.2.154.82 - 3Com SuperStack
Interfaces
1687 - RMON 10Mbps Se
1904 - Port 24 on Unit 1
5648 - Port 5 on Unit 1
30613 - Management Port
40039 - Port 1 on Unit 1
44410 - Port 6 on Unit 1
48765 - Port 4 on Unit 1
57057 - Port 2 on Unit 1
64566 - RMON 100Mbps Segment on Uni
Ethernet Stats
TopN(5)
Agent: 1/ 1
ethernet-csmacd
Speed: 10 Mb
Start
Explorer - Papierkorb
Microsoft Word - Unicenter...
RMON Analysis
14:15

Abbildung 01-4: RMON Analyse

Etwas problematisch sind zum einen der hohe Zeitbedarf für die Generierung der Berichte aus den Datenbeständen und der hohe Plattenplatzbedarf. Für den Management- und Repositoryserver sollte eine performante Maschine mit entsprechendem Hauptspeicher- und Plattenausbau benutzt werden. Dies gilt allerdings grundsätzlich für praktisch alle Managementwerkzeuge, die mit hohem Daten- und Auswerteaufkommen umgehen müssen. Gerade in der Detailanalyse von High-Speed-Netzen treten naturgemäß in kurzer Zeit hohe Datenmengen auf.

Unangenehm aufgefallen ist die Tatsache, dass beim Generieren von Berichten eventuelle Ausfallzeiten der Probes nicht gesondert gekennzeichnet werden. Werden Service Level

Agreements mit Hilfe der Probes überwacht, ist zu gewährleisten, dass alle notwendigen Informationen für den Betrachtungszeitraum zur Verfügung stehen.

KAPITEL 02 HP TOOLS

HEINER RISTHAUS, RONALD SCHATEN

02.01 Einleitung

HP bietet mit MeasureWare und PerfView innerhalb der RPM (Resource and Performance Management) Tools zwei Produkte für die kontinuierliche Erfassung, Überwachung und Auswertung von Performancedaten. Weitere Produkte sind z.B. NetMetrix mit dem Fokus auf Hardware(RMON)daten und GlancePlus mit dem Fokus auf Echtzeitbetrachtungen ("Was ist in diesem Augenblick gerade im Netz los?"). Hier wird über Erfahrungen aus Evaluierungen mit einigen der Werkzeuge berichtet. Weitere Informationen sind bei www.hp.com erhältlich.

02.02 HP MeasureWare/PerfView

Hier werden die beiden HP-Produkte behandelt, die sich speziell dem Bereich Resource Management bzw. Performance Management widmen..

MeasureWare und PerfView bilden gemeinsam oder in Verbindung mit zusätzlichen Komponenten ein Manager/Agenten-Konzept. Mit MeasureWare ist die Agententechnologie bezeichnet, mit PerfView ein dafür speziell entwickeltes Auswertetool. Als zusätzliche Managementkomponenten können prinzipiell beliebige Netz- und Systemmanagementsysteme (z.B. Visualisierung von Alarmen oder Problemzuständen) zum Einsatz kommen.

(a) MeasureWare

MeasureWare ist eine Agententechnologie, die für das Sammeln von Performancedaten von Betriebssystemen, Applikationen, Datenbanken etc. zuständig ist. Über eine grafische Oberfläche kann die Erfassung parametriert werden. Dazu gehören Häufigkeit, Menge und Typ der Daten und die Definition von Schwellwerten und Alarmen.

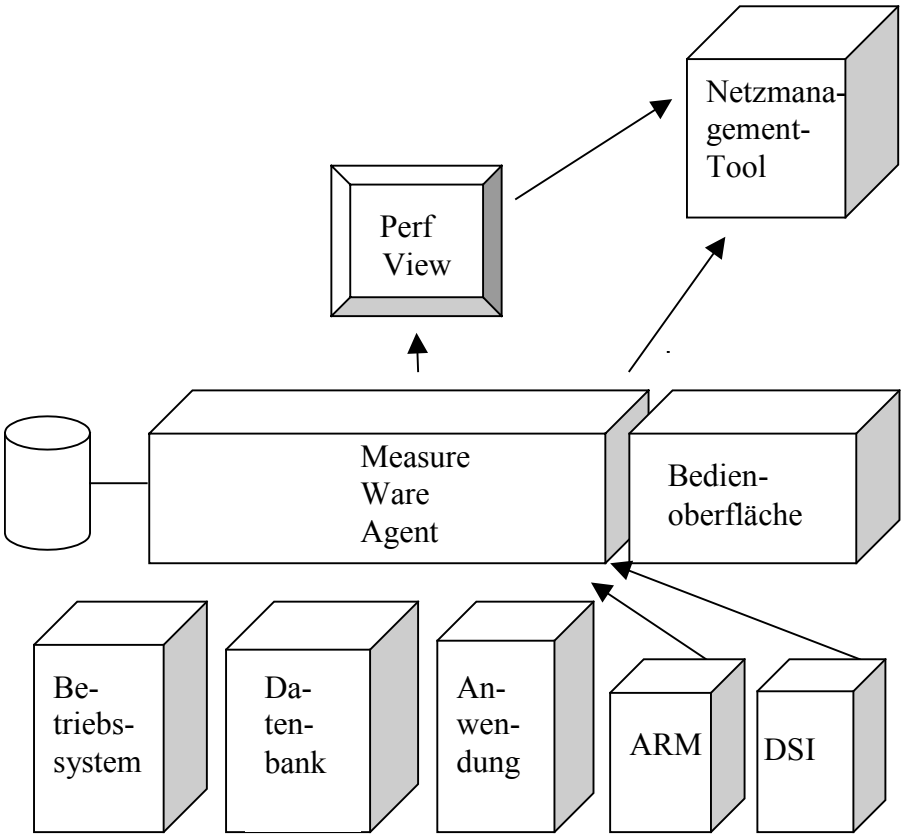


Abbildung 02-1: HP PerfView / MeasureWare Architektur

Verfügbar ist MeasureWare für Windows und Unix-Derivate. Außerdem gibt es Agenten für Anwendungen und Datenbanken (SAP R/3, Oracle, Informix, ...). Alarme können grundsätzlich an beliebige Netzmanagementwerkzeuge wie HP OpenView PerfView, HP Network Node Manager oder Systeme von Fremdanbietern weitergeleitet werden.

Der MeasureWare-Agent wird auf den verteilten und zu vermessenden Systemen installiert und erfasst lokal Performancedaten. Messwerte werden mit Timestamps versehen und gespeichert. Es kann durch Konfiguration der Agenten eine lokale Verdichtung der erfassten Performancedaten vorgenommen werden. Der Agent bietet eine ARM Schnittstelle, sodass entsprechend instrumentierte Applikationen Informationen über das Zeitverhalten ihrer Geschäftstransaktion an MeasureWare liefern können. DSI (data source integration) ermöglicht die Einbringung von Informationen durch Applikationen, die solche Daten z.B. in

Logfiles schreiben. Die Logfiles können ausgewertet und die Daten nach MeasureWare übernommen werden..

(b) PerfView

PerfView erledigt die Auswertung, Analyse und Verdichtung der Daten, die durch den MeasureWare-Agenten geliefert werden. Historische Daten werden per Graph angezeigt. Diagramme sind parametrierbar. Durch geeignete Parametrisierung von Auswertungen sind Vergleiche verschiedener Systeme möglich, ebenso eine kontinuierliche Parameterüberwachung im Sinne von Service Level Überwachung. Es gibt eine Historienliste aufgetretener Alarme, durch Doppelklicken auf einen Alarm wird automatisch ein Diagramm geöffnet, das die Situation grafisch darstellt. Auf Basis bereits erfasster Daten kann auch nachträglich eine Schwellwertanalyse durchgeführt werden. D.h., es wird ausgegeben, welche Alarme erzeugt worden wären, hätte man vorher die Schwellwerte auf einen bestimmten Wert gesetzt. Damit kann überprüft werden, wie gut eine Parametrierung in der Vergangenheit gepasst hätte. PerfView eignet sich besonders für die Erkennung von Trends, Problemen und Engpässen im System.

(c) Sonstiges

Die Tools sind leicht und schnell zu installieren. Wie alle funktional umfangreichen Werkzeuge im Netz- und Systemmanagement neigen auch die HP Tools dazu, den Benutzer mit Informationen und bunten Grafiken zu überfrachten und unübersichtlich zu sein. Die sinnvolle Parametrierung und Konfiguration des Gesamtsystems ist aufwändig.

Während einfache Parametrierungen und Bedienvorgänge über die grafische Oberfläche erledigt werden können (z.B. start collection, stop collection, Dateiim- und -export, ...), müssen komplexe Funktionen über Skripte und Parametrierungsdateien definiert werden (Schwellwerte, Alarmziele, Korrelationsregeln etc.). Dazu existiert eine eigene Quasi-Programmiersprache.

Die verteilte Haltung der Performancedaten verringert die benötigte Server(Manager)kapazität, außerdem ergibt sich daraus eine relativ geringe Netzbelastung.

Die grafische Aufbereitung der historischen Daten erlaubt fast beliebige Zoomfaktoren. Zudem ist es möglich, für ein Service Level Management geeignete Reports zu generieren, um eine Übersicht z.B. über Schwellwertüberschreitungen zu bekommen. Prognostische Funktionen für die Netz- und Kapazitätsmodellierung fehlen, in diesem Sinne sind die Tools reine Analysewerkzeuge.

02.03 HP Ethernet LanProbe / HP NetMetrix

(a) Die Hardware

Bei der HP Ethernet LanProbe handelt es sich um dedizierte Hardware, die an bestehende Netzsegmente angeschlossen werden kann und dort den Verkehr mitschneidet.

Die Basiskonfiguration der LanProbe wird über einen seriellen Anschluss und durch einen Arbeitsplatzrechner vorgenommen. So können menügesteuert Einstellungen für den späteren Betrieb gemacht werden, beispielsweise die Netzwerkeinstellungen, unter denen die Probe später im Netz angesprochen werden soll, Systemdatum und Uhrzeit, oder die Konfiguration der seriellen Schnittstelle.

Die weitere Konfiguration kann per NetMetrix über das Netzwerk vorgenommen werden. Die Probe arbeitet dann völlig unbeaufsichtigt. Die von der LanProbe gesammelten Daten können über das Netzwerk abgefragt oder über ein direkt an die Probe angeschlossenes Modem an einen entfernten Rechner weitergegeben werden. Natürlich können die Daten auch direkt per seriellem Kabel an einen Rechner weitergegeben werden. Die durch die Probe unterstützten Software-Standards sind

- Remote Network Monitoring Management Information Base (RFC 1757)
- SNMP MIB-II (RFC 1213)
- SNMP (RFC 1157)
- RMON-2 (RFC 2021)
- Remote Network Monitoring MIB Protocol Identifiers (RFC 2074)
- HP LanProbe Private MIB

(b) Die Software

Softwareseitig ist HP OpenView NetMetrix/UX in der Version 6 auf einem Rechner mit HP-UX 10 zum Einsatz gekommen. Bei der Installation gab es keine größeren Probleme. Für das Sammeln der Daten muss ein spezieller Dämon gestartet werden. Der Agent Manager dient zur Verwaltung der verschiedenen Datenquellen. Mit ihm können die Agenten eingerichtet und konfiguriert werden, und von hier aus können die anderen Diagnose-Tools gestartet werden. Es können auch verschiedene Graphen generiert werden, um die aktuelle Performance des Netzes analysieren und beurteilen zu können.

Zunächst ist es notwendig, die Agenten einzurichten, von denen man die Daten beziehen möchte. Dazu benötigt man lediglich die Netzwerkadresse und die Community der jeweiligen Agenten. Innerhalb der Benutzerumgebung werden diese Daten dann in einer Art

Baumstruktur angezeigt. Hier kann man übersichtlich die Datenquelle auswählen, mit der man aktuell arbeiten möchte.

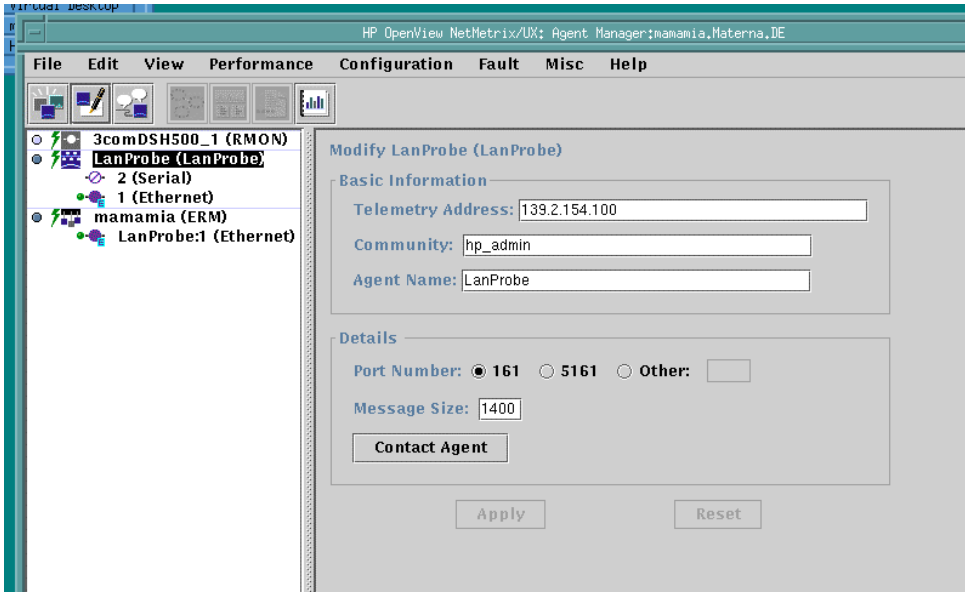


Abbildung 02-2: Der Agent Manager von HP Open View NetMetrix

Wenn man beabsichtigt, die Extended-RMON-Fähigkeiten der LanProbe zu benutzen, dann muss auch ein ERM-Agent (Extended RMON Datenquelle) angegeben werden, mit dem die LanProbe dann assoziiert werden kann.

Durch einen Klick auf eine Datenquelle werden rechts im Fenster die dazugehörigen Konfigurationseinstellungen angezeigt und die aktuellen Einstellungen können verändert werden (Portnummer, welche Daten sollen in welchen Abständen gesammelt werden, mit welchem Agenten soll die Probe assoziiert werden), ...

(c) Protocol Analyzer

Mit dem Protocol Analyzer ist es möglich, nach bestimmten Filterregeln Datenpakete im Netzwerk zu sammeln und auszuwerten. Es ist möglich, mehrere Instanzen dieser Auswertungen gleichzeitig zu betreiben. Die Filterung und die Speicherung der Datenpakete übernimmt dabei die LanProbe selbst. Mit dem Analyzer kann man lediglich die Regeln festlegen und die gesammelten Daten auswerten.

Es besteht die Möglichkeit, nach Konversationen zu filtern, an denen bestimmte Hosts beteiligt sind, oder nach Paketen, die bestimmte Inhalte enthalten. So kann man zum Beispiel alle Pakete sammeln, die zwischen zwei bestimmten Rechnern ausgetauscht werden und Daten über X11 enthalten. Man kann bis zu acht Suchmuster definieren, bei denen jeweils nach bestimmten Zeichen an einer festgelegten Stelle im Paket gesucht wird. Es ist auch möglich, verschiedene Filter zu kombinieren, die dann logisch verknüpft werden können.

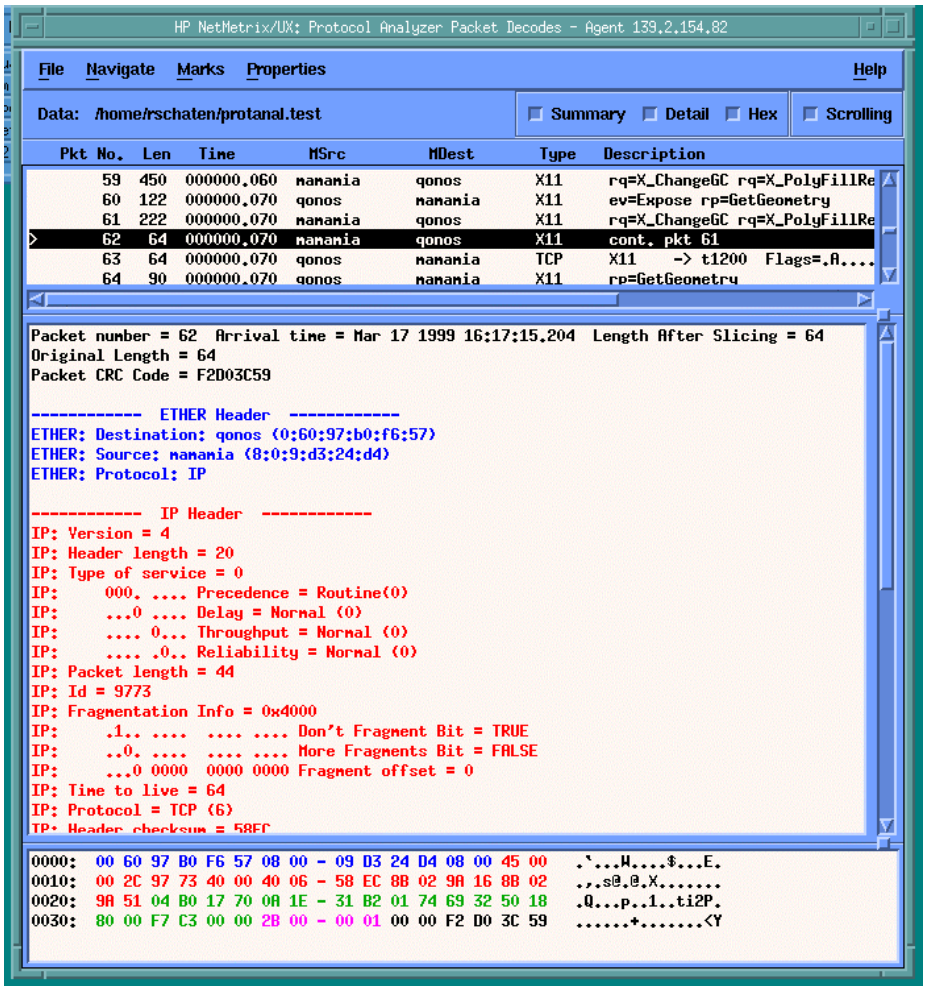

HP NetMetrix/UX: Protocol Analyzer Packet Decodes - Agent 139.2.154.82
File Navigate Marks Properties Help
Data: /home/rschaten/protanal.test
Summary Detail Hex Scrolling
Pkt No. Len Time MSrc MDest Type Description
59 450 000000.060 mamamia qonos X11 rq=X_ChangeGC rq=X_PolyFillRe
60 122 000000.070 qonos mamamia X11 ev=Expose rp=GetGeometry
61 222 000000.070 mamamia qonos X11 rq=X_ChangeGC rq=X_PolyFillRe
62 64 000000.070 mamamia qonos X11 cont, pkt 61
63 64 000000.070 qonos mamamia TCP X11 -> t1200 Flags=.A....
64 90 000000.070 qonos mamamia X11 rp=GetGeometry
Packet number = 62 Arrival time = Mar 17 1999 16:17:15.204 Length After Slicing = 64
Original Length = 64
Packet CRC Code = F2D03C59
---------- ETHER Header ----------
ETHER: Destination: qonos (0:60:97:b0:f6:57)
ETHER: Source: mamamia (8:0:9:d3:24:d4)
ETHER: Protocol: IP
---------- IP Header ----------
IP: Version = 4
IP: Header length = 20
IP: Type of service = 0
IP: 000. .... Precedence = Routine(0)
IP: ...0 .... Delay = Normal (0)
IP: .... 0... Throughput = Normal (0)
IP: .... .0.. Reliability = Normal (0)
IP: Packet length = 44
IP: Id = 9773
IP: Fragmentation Info = 0x4000
IP: .1.. .... Don't Fragment Bit = TRUE
IP: ..0. .... More Fragments Bit = FALSE
IP: ...0 0000 0000 0000 Fragment offset = 0
IP: Time to live = 64
IP: Protocol = TCP (6)
0000: 00 60 97 B0 F6 57 08 00 - 09 D3 24 D4 08 00 45 00 .`...W....$...E.
0010: 00 2C 97 73 40 00 40 06 - 58 EC 8B 02 9A 16 8B 02 .,.s@.@.X.......
0020: 9A 51 04 B0 17 70 0A 1E - 31 B2 01 74 69 32 50 18 .Q...p..1..ti2P.
0030: 80 00 F7 C3 00 00 2B 00 - 00 01 00 00 F2 D0 3C 59 ......+.......<Y

Abbildung 02-3: Der Protocol Analyzer von HP Open View NetMetrix

Die gesammelten Pakete kann man sich dann mit Hilfe des View-Tools aus dem Packet Analyzer ansehen. Hier sieht man in einem dreigeteilten Fenster alle Pakete, die die LanProbe gesammelt hat, vgl. Abbildung 02-3. Neben der laufenden Nummer und der Länge des Paketes sieht man hier auch einen Zeitindex. Dieser zeigt die Zeit entweder relativ zum ersten Paket, relativ zum vorhergehenden Paket oder absolut nach der internen Uhr der LanProbe an. Außerdem werden Quell- und Zielrechner, Pakettyp und eine Kurzbeschreibung angezeigt.

In dem mittleren Fenster sieht man eine detaillierte Darstellung des selektierten Pakets. Hier wird nach den verschiedenen Protokollschichten aufgeteilt, und das Feld liefert sehr tiefgehende Informationen über den Inhalt. Im letzten Feld sieht man das Paket in einer hexadezimalen und in einer ASCII-Darstellung. Auch hier sind die verschiedenen Protokollebenen farbig markiert. Wenn man im mittleren oder im unteren Feld einen Teil selektiert, wird im jeweils anderen Feld angezeigt, welchen Teil des Paketes man ausgewählt hat. So kann man sich ein genaues Bild über die Inhalte der gesammelten Pakete machen

(d) Report Generator

Mit Hilfe des Report Generators kann man sich Auskunft über den Zustand des Netzwerkes innerhalb eines bestimmten Zeitraumes verschaffen. Nach dem Klick auf das entsprechende Icon öffnet sich zunächst das Report Status Fenster. Von hier aus kann man neue Reports anlegen, die Generierung der Seiten anstoßen und sich das Ergebnis anzeigen lassen. Hier sieht man auch das Scheduling und das Ausgabeziel der einzelnen Reports, vgl. Abbildung 02-4.

Von hier aus gelangt man in den eigentlichen Report Generator. Hier kann man festlegen, welche Daten angezeigt werden sollen, und aus welchen Quellen diese stammen sollen.

Selbstverständlich ist konfigurierbar, welche Daten dargestellt werden sollen, wie die Daten in dem Graphen dargestellt werden sollen, wie die Beschriftung des Graphen aussehen soll, welche Art von Graph erstellt werden soll (Torten-, Balkendiagramme etc.) und ähnliches. Einem Report können Graphen verschiedenen Typs hinzugefügt werden.

Die Erstellung der Reports kann zusätzlich scheduled werden, d.h. zu bestimmten Zeitpunkten einmalig oder periodisch veranlasst werden. Als Ausgabemedium kann zwischen Bildschirmausgabe, Druckausgabe, Ausgabe als Postscript-Datei, Ausgabe per Mail oder Ausgabe an ein benutzerdefiniertes Kommando gewählt werden. Oben in der Grafik sehen wir, dass der Zustand des Netzes über einen bestimmten Zeitraum dargestellt wird. Es wird die prozentuale Auslastung und die totale Anzahl der entdeckten Fehler dargestellt.

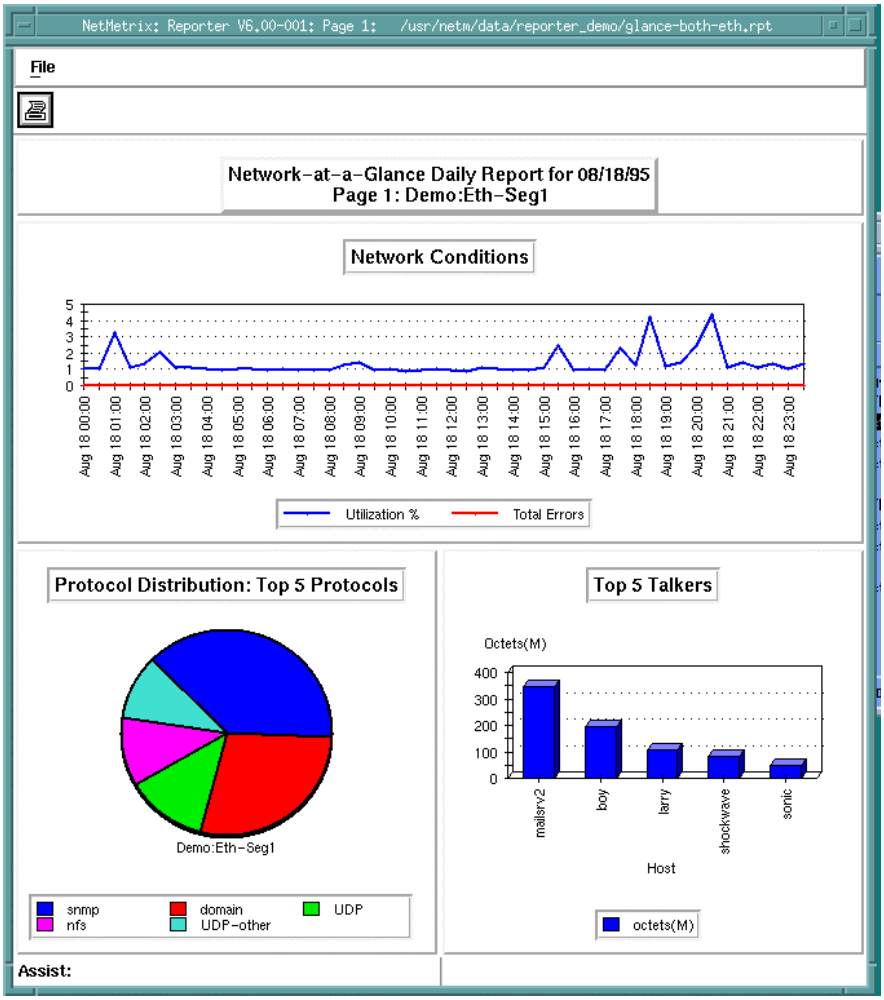


Abbildung 02-4: Der Reporter von HP/NetMetrix

Unten links kann man erkennen, auf welche Protokolle sich die gemessene Netzlast im überwachten Zeitraum verteilt. Es werden nur die fünf häufigsten Protokolle angezeigt. Unten rechts sieht man dann die fünf Top-Talker, also die fünf Rechner, die den meisten Netzwerkverkehr verursacht haben.

(e) Abschließende Bewertung bzgl. der Eignung als Performanceanalysetool

Im konkreten Fall ging es darum, die Antwortzeit einer Datenbankanwendung zu bewerten, d.h. die Zeit zwischen dem Tastendruck des Benutzers und dem Erscheinen der entsprechenden Daten auf dem Benutzerbildschirm. Näherungsweise sollten die Daten auf dem Netzwerk gemessen werden, also die Zeit, die das Client-Programm der Datenbank braucht, um das Antwort-Datenpaket zusammenzustellen.

Mit der LanProbe kann man zwar die Zeitpunkte feststellen, zu denen die Pakete auf dem Netzwerk erscheinen, allerdings bezogen auf die interne Uhr der Probe. Für genauere Betrachtungen müssen diese Zeiten mit denjenigen innerhalb der Datenbank oder der Clientanwendung korreliert werden. Dieses erfordert, dass die Uhr ausreichend genau mit denen der beteiligten Rechner synchronisiert wird, um Messungen in diesen Zeitbereichen (wenige Millisekunden) sinnvoll vorzunehmen. Dazu müssten spezielle Maßnahmen ergriffen werden, die Eingriffe in die Applikation erfordern. Die LanProbe unterstützt solche Forderungen von sich aus nicht.

Außerdem bietet sie keine Möglichkeit der Modellierung von "Beginn" und "Ende" von Anfragen bzw. Antworten. Damit muss die Auswertung letztlich manuell erfolgen und ist dementsprechend aufwändig. Die Probe wird dann sinnvoll eingesetzt, wenn es darum geht, Statistiken über die Netzwerkauslastung zu erhalten. Mit ihr lassen sich Auslastung, Protokollverteilung, Top-Talker, beschädigte Datenpakete etc. komfortabel aufzeichnen und visualisieren.

Damit hat LanProbe ihren Schwerpunkt im diagnostischen Bereich der Fehleranalyse und im Bereich der frühzeitigen (proaktiven) Problemvermeidung.

KAPITEL 03 GLANCEPLUS

JOSEF BERSCHNEIDER.

GlancePlus ist ein System Performance-Monitor und Diagnose-Tool. Das Programm gehört zu den HP OpenView Ressourcen- und Performance-Produkten. Es ist ein Online Performance-Monitor, mit dem eventuell vorhandene Systemengpässe angezeigt und die Verursacher (Prozesse) ermittelt werden können. GlancePlus ist für die Plattformen HP-UX, Solaris, AIX und Sinix verfügbar. Es wurde nur auf HP-Systemen für die Erprobung betrieben. Ob alle Funktionen und Messwerte auf allen Plattformen zur Verfügung stehen, kann deshalb nicht bewertet werden.

Gute Performance-Kenntnisse vorausgesetzt, ist das Tool leicht zu erlernen und zu bedienen. Nachfolgend werden die Funktionen und Schnittstellen kurz beschrieben.

03.01 Benutzerschnittstelle

Es stehen zwei Benutzerschnittstellen zur Verfügung:

@ Die Character-Schnittstelle, die mit dem Programm *glance* aktiviert wird. Damit können die Messdaten auch ohne X-Windows dargestellt werden. Die Steuerung erfolgt im Kommandomodus.

@ Die Grafische Schnittstelle (Motif-based) wird mit dem Programm *gpm* gestartet. Mit dieser Schnittstelle steht auch eine kurze Historie der Messdaten, abhängig von der Intervall-Länge, zur Verfügung. Die Diagrammform (Balken, Kreis und Fläche) ist wählbar. Die Steuerung erfolgt per Mouse-Click.

Bis auf die Bedienung und die Darstellung der Daten sind die angezeigten Messwerte im Wesentlichen identisch.

03.02 Hierarchische Darstellung (Top-Down Analyse)

Es stehen über 1000 Messwerte zur Verfügung. Im Hauptschirm werden folgende Komponenten dargestellt:

@ CPU (Auslastung in %, Anteile sind getrennt nach user und system)

@ Memory (Pagingrate für Ein-/Ausgabe)

- Disk (getrennt nach user, system, virtual memory und raw I/Os)
- Network (Anzahl Pakete für Ein-/Ausgabe)

Die Darstellung jeder dieser Komponenten ist als Graph und Button realisiert. Im Graph werden die Messwerte über die Zeit angezeigt. Die definierten Schwellwerte werden durch die Farbe des Buttons (Ampelfarben) signalisiert. Ausgehend von dieser Summary-Übersicht kann man sich im Top-Down Verfahren sehr detaillierte Messdaten anzeigen lassen.

03.03 Definition von Applikationen

Durch die Levels System, Applikation und Prozesse hat man eine unterschiedliche Sicht auf die Systemperformance. Durch die Zusammenfassung von Prozessen zu Applikationen hat man eine zusätzliche Sicht zwischen den globalen Systemdaten und den spezifischen Messdaten pro Prozess. Die Zusammenfassung von Applikationen kann auf Basis von User-, Group- und Prozessnamen erfolgen. Der Prozessname ist auf 12 Zeichen begrenzt. In der Erpobungsphase gab es Probleme, Prozesse den vorhandenen Applikationen zuzuordnen, wenn sie sich nur in den Parametern der Kommandos unterscheiden (z.B. ./utmwork apl1 x1 ..).

03.04 Funktionen von GlancePlus

(a) Sortierung, Anordnung, Filter und Intervalle

Durch Sortierung und Anordnung der Messwerte ist eine individuelle Sicht definierbar. Auf Basis von Grenzwerten können die anzuzeigenden Daten gefiltert werden. Die Anzeige der Messdaten kann in Intervallen von 2 bis 32767 Sekunden variiert werden.

(b) Alarmfunktion

Werden Engpässe festgestellt, die über Schwellwerte definiert werden können, erfolgt über die Alarmfunktion eine optische Warnung. An einen Alarm kann zusätzlich eine Aktion gekoppelt sein, um auf einen Engpass automatisch, per Programm oder einer Kommandofolge zu reagieren.

(c) Hilfefunktion

Die Online Hilfefunktion bietet sowohl Bedienungshinweise als auch für jeden Messwert eine kurze Erklärung.

(d) Adviser-Modus

Mit dem adviser_only Modus werden Messdaten in eine Datei geschrieben, ohne dass die Bildschirmausgabe aktiviert wird. Die Messwerte können über eine Kommandoschnittstelle individuell ausgewählt werden. Damit können auch Langzeitstatistiken erstellt und die Daten mit anderen Programmen (z.B. EXCEL) ausgewertet und dargestellt werden. Die Schnittstelle dient auch zur Übergabe der Messdaten an die HP OpenView Produkte, wie MeasureWare und PerfView.

(e) Überwachung von Transaktionen

Es besteht die Möglichkeit, sich Messdaten wie

- mittlere Transaktionszeit und Verteilung der Transaktionszeit
- Anzahl Transaktionen
- Einhaltung von Service Levels (z.B. ist die Transaktionszeit kleiner 2 Sekunden)

vom MeasureWare Agent Application Responce Measurement (ARM) anzeigen zu lassen. Die Applikationen müssen dazu mit dem Transaktion Tracker vorbereitet sein. Der Transaktion Tracker ist Bestandteil des Application Programm Interfaces (API), mit dem der Programmierer Transaktionen innerhalb von Applikationen definieren kann.

03.05 Abgrenzung zu den Standard-Messtools in UNIX

Glance benutzt spezielle Trace-Funktionen im HP-UX Kernel und setzt diese in Messwerte über den midaemon Prozess um. Somit können wesentlich genauere und zusätzliche Messdaten gegenüber den Standardtools wie sar, vmstat und iostat zur Verfügung gestellt werden. Diese „kmem“ Standardtools ermitteln ihre Daten aus Zählern im Kernel über ein Sampling Verfahren.

Mit Glance stehen alle Messwerte der einzelnen Tools über eine Schnittstelle zur Verfügung. Hervorzuheben sind die Messwerte, die für die einzelnen Prozesse zur Verfügung gestellt werden.

Es stehen folgende wesentlichen Messwerte pro Prozess zur Verfügung:

- CPU-Verbrauch (user-, system-, systemcall-, interrupt- und cswitch-Anteil)
- I/O-Rate, -Zeit, -Byte (Disk, Filesystem)
- Memory (resident, virtual, shared)
- Systemcalls

@ Sonstige Messwerte (geöffnete Dateien, IPC, NFS, DCE, MSG, RPC, STOP-Reason)

03.06 Zusammenfassung

Mit GlancePlus steht ein sehr gutes Tool für Performance Analysen zur Verfügung. Mit den über 1000 Messpunkten ist eine sehr detaillierte Diagnose bei System Performance-Problemen möglich. Nachteilig ist, dass standardmäßig in der LAN-Statistik keine Messwerte für das Nachrichtenvolumen (Anzahl übertragener Bytes) vorhanden sind. Hervorzuheben ist die sehr nützliche Online Erklärung zu den einzelnen Messwerten.

KAPITEL 04 BEST/1[1] – EINE ÜBERSICHT

JÜRGEN LÜHRS

04.01 Betrieb

Best/1 basiert auf Agenten (Collectors), die auf den zu überwachenden Rechnern (im Folgenden auch Knoten genannt) installiert werden müssen. Diese Agenten sammeln System-Daten und Daten über Datenbanken. Ein Performance Modul für SAP R/3 ist ebenfalls verfügbar. Ein Knoten ist der sog. „managing node". Auf ihm werden zusätzlich die Komponenten zur Auswertung aller von den Agenten gesammelten Daten installiert. Alle anderen Knoten erhalten nur die Agenten und heißen „managed nodes". Die „managed nodes" übertragen die auf ihnen erhobenen Daten zur Auswertung auf den „managing node".

04.02 Die Komponenten von BEST/1

Die Komponenten Collect, Monitor, Manager, Analyze und Predict stehen auf dem „managing node" zur Verfügung. Sie sind nur für Unix implementiert und bieten eine GUI über ein X-Terminal. Die Komponente Visualizer arbeitet nur unter Windows. Die Komponenten werden im Folgenden beschrieben.

(a) Collect

Collect erhebt und speichert Kernel-Level Performance Daten. Sie werden in Real-Zeit auf den einzelnen Knoten ermittelt. Diese Daten dienen zum

- Real time monitoring (in der Komponente Monitor)
- „What if...?" modeling (in den Komponenten Analyze und Predict)
- Darstellen historischer Daten (in der Komponente Visualizer).

Die von Collect erhobenen Daten enthalten Metriken für

---

[1] Die Software BEST/1 ist ein Werkzeug zur Kapazitätsplanung und zum Performancemanagement. BEST/1 wurde ursprünglich von der Firma BGS entwickelt. BGS wurde von der Firma BMC aufgekauft, welche das Produkt weiter entwickelt und vertreibt. BEST/1 ist für den Mainframe-Bereich, UNIX und Windows-NT-Umgebungen verfügbar.

- CPU, Memory, Prozesse
- Disks, Logische Laufwerke, I/O activity
- Datenbanken (Oracle, Sybase, Informix)

Es gibt System-Collectoren für UNIX- und Windows NT-Daten, Collectoren für die Datenbanken von Oracle, Sybase und Informix sowie auf SNMP- und ARM-basierende Collectoren. Ein Collector UMX sammelt Daten von externen Quellen. Über das Collect-GUI können die Knoten, von denen Daten erhoben werden sollen, angegeben werden. Die Daten werden nach dem Ende der Erhebung auf den „managing node“ übertragen.

(b) Visualizer

Die Komponente Visualizer dient der Darstellung der Daten, die von den Komponenten Analyze und Predict erzeugt wurden. Sie können in graphischer Form oder als Report angezeigt werden.

(c) Manager

Die Komponente Manager automatisiert die Datenerhebung, Datenverarbeitung und das Datenmanagement im BEST/1-System. So kann über einen Scheduler der Zeitraum, über den die Daten erhoben werden sollen, eingestellt werden. Ebenso kann die Weiterverarbeitung der Daten in den Komponenten Analyze und Predict sowie die Erzeugung von Visualizer-Files konfiguriert werden. Die erzeugten Visualizer-Files können mittels eines auf FTP-basierenden Skripts auf den Visualizer-PC übertragen werden. Weiterhin besteht die Möglichkeit, früher erhobene Daten zu komprimieren, zu archivieren oder ab einem einstellbaren Alter zu löschen.

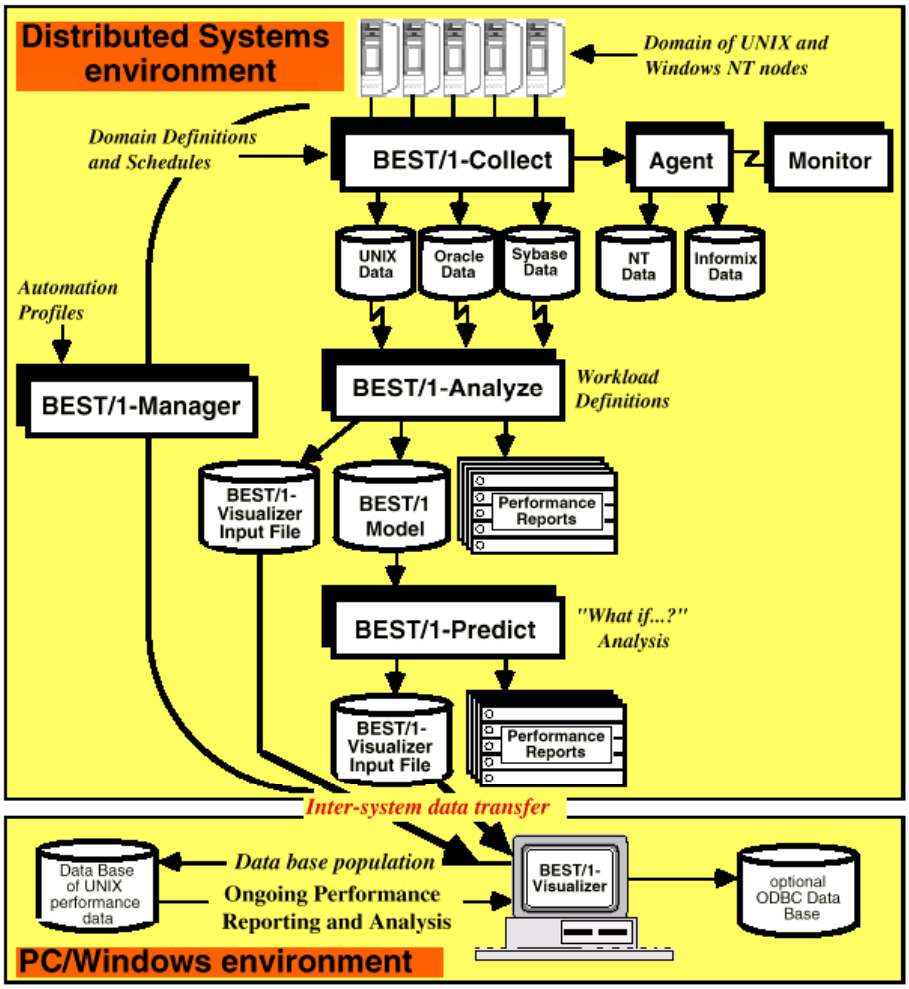


Abbildung 04-1: Die Komponenten von BEST/1

(d) Monitor

Die Komponente Monitor bietet eine Realzeit-Darstellung der von Collect erhobenen Daten in graphischer Form. Weiterhin lässt sich eine Schwellwertüberwachung ausgewählter Parameter konfigurieren. Für den Fall einer Schwellwertüberschreitung können folgende „Actions“ ausgelöst werden:

- @ Generieren eines SNMP Traps
- @ Versenden einer Mail (z. B. an den zuständigen Administrator)
- @ Starten eines Skripts (Fehler Recovery)

(e) Analyze

Die Komponente Analyze erzeugt aus den von Collect erhobenen System- und Datenbankdaten Performance Reports und ein Modell, das der Komponente Predict als Eingabe dient. Das Modell repräsentiert

- @ die Performance Charakteristika, die von der Komponente Collect in einem spezifizierten Zeitintervall ermittelt wurden.
- @ die Workloads, die mit dem Analyze Workload-Charakterisierungs-Tool definiert wurden. In BEST/1 besteht ein Workload aus Transaktionsklassen und Benutzerklassen. In einer Transaktionsklasse werden die Prozesse der zu untersuchenden Anwendung eingetragen (z. B. alle Prozesse einer Datenbank). In einer Benutzerklasse werden die Benutzer der zu untersuchenden Anwendung aufgenommen (z. B. die Benutzer der Datenbank). Die aus einer Transaktionsklasse und einer Benutzerklasse gebildete Workload liefert nach der Analyse mit Analyze Informationen darüber, wer die Arbeit auf einem System verrichtet (spezifiziert durch die Benutzerklasse) und welche Arbeit auf einem System verrichtet wird (spezifiziert durch die Transaktionsklasse).
- @ Client/Server Analysen über mehrere Knoten. In BEST/1 können die Transaktionsklassen in Abhängigkeit voneinander definiert werden, so dass Client/Server-Systeme abgebildet werden können. Hierbei wird (werden) der (die) Clientprozess(e) eines Client/Server-Systems in einer „independent transaction“, und der (die) Serverprozess(e) in einer „dependent transaction“ definiert. Client- und Serverprozess(e) können hierbei auf einer oder auf verschiedenen Maschinen laufen. Die Plattformen brauchen nicht identisch zu sein (funktioniert z. B. mit NT als Clienten- und UNIX als Server-Plattform). Es kann allerdings nur eine Abhängigkeitsebene modelliert werden. Die Struktur Client – Applicationserver – Datenbankserver lässt sich in BEST/1 nicht modellieren, da hierbei zwei Abhängigkeitsebenen gefordert werden.
- @ vordefinierte Transaktionen, die I/O- und Netzwerk-Aktivitäten repräsentieren.

Weiterhin kann Analyze die von der Komponente Visualizer benötigten Dateien erzeugen.

(f) Predict

Die Komponente Predict verwendet das Warteschlangen-Netzwerk-Modell, das in Analyze gebildet wurde, berechnet es und erzeugt Reports, die die Performance des betrachteten Systems wiedergeben. In Predict lassen sich Schwellwerte für Service Levels setzen und

auf Einhaltung überprüfen. Durch Modifikation der Modell-Parameter lassen sich Performance Vorhersagen treffen über

- steigende Service Anforderungen durch Hinzufügen von neuen Benutzern und Applikationen.
- Änderung der Konfigurationen zur effizienteren Nutzung existierenden Equipments durch Verschieben von Workloads, Transaktionen oder Benutzer.
- Hardware-Änderungen (CPU- oder Disk-Aufrüstung).
- Wachstum der Auslastung durch angenommene Geschäftsexpansion.

Bevor Modifikationen am Modell vorgenommen werden, wird das Modell kalibriert und danach ein Baselining des Modells durchgeführt. Das Kalibrieren stellt sicher, dass das Modell ein getreues Abbild des zu untersuchenden Systems ist. Das Kalibrieren erfolgt durch Vergleich der von Predict berechneten Werte für CPU Throughput, CPU Utilization und CPU Queue Length mit den von Analyze gemessenen Werten. Liegen diese nicht innerhalb einer spezifizierten Abweichung, müssen sie im Modell von Hand korrigiert werden. Hierbei spielt manchmal das Netzwerk- und Memory-I/O-Verhalten der Applikationen eine große Rolle. Diese Daten kann BEST/1 manchmal nicht exakt genug ermitteln. Sie müssen auf andere Weise bestimmt und in BEST/1 eingegeben werden. Dies kann mit erheblichem Aufwand verbunden sein. Eine exakte Kalibrierung ist von großer Bedeutung für die Genauigkeit der berechneten Vorhersagen.

Das „Baselined Model“ repräsentiert den aktuellen Zustand des zu untersuchenden Systems nach der Kalibrierung. Auf dieser Basis können nun Performance Vorhersagen durch Modifikation der Modell-Parameter vorgenommen werden.

Predict berechnet eine Antwortzeit bezogen auf die zugrunde liegenden Transaktionen. Auch die Genauigkeit dieser Berechnung ist natürlich von der gelungenen Kalibrierung abhängig. So müssen exakte Informationen über CPU- und I/O-Bedarf der Transaktion eingegeben werden. Die genaue Vorgehensweise hierzu ist noch Gegenstand der Untersuchung. Die Antwortzeit schlüsselt sich auf in

- CPU Service: durchschnittliche Antwortzeit verursacht durch die CPU-Zeit der (Client-)Transaction.
- CPU Wait: durchschnittliche Antwortzeit verursacht durch Warten der Transaction auf einen Service (queuing).
- IO Service: Antwortzeit des I/O Systems.
- Other Service und Other Wait: gesamte Antwortzeit eines Client/Server-Systems.

Handelt es sich um ein Client/Server-System, wird die Server-Komponente ebenfalls in CPU-Zeit und I/O-Zeit aufgelöst. Weiterhin kann Predict die von der Komponente Visualizer benötigten Dateien erzeugen.

04.03 Aufbau eines BEST/1-Modells

Abbildung 04-2 veranschaulicht die Komponenten eines BEST/1-Modells.

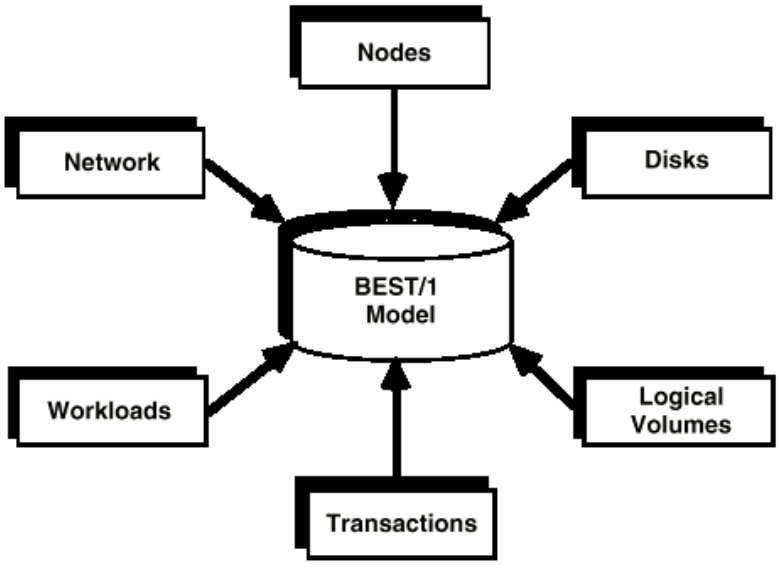


Abbildung 04-2: Aufbau eines BEST/1-Modells

(a) Network

Ein Network repräsentiert eine logische Gruppe von Knoten, die über ein LAN (z.B. Ethernet) verbunden sind. Die Knoten können UNIX- oder Windows NT-Systeme sein. Es wird nur ein Netzwerktyp pro Modell unterstützt. Der prozentuale Netzwerkverkehr zu Knoten außerhalb des Modells kann eingegeben werden. Da die BEST/1 Collectoren den Netzwerkverkehr nicht untersuchen, müssen diese Zahlen mit anderen Werkzeugen ermittelt werden.

(b) Nodes

Nodes repräsentieren die CPUs im Netzwerk. Wichtige Node Attribute sind CPU Typ und Memory Größe. Predict berechnet den prozentualen Anteil der Zeit, in der die CPU beschäftigt ist und die CPU Queue Length.

(c) Transaction

Eine Transaction repräsentiert die kleinste Einheit der Arbeit auf einem System. Sie kann andere Transactionen auslösen (Client/Server).

(d) Workload

Eine Workload fasst definierte Transactionen und die Benutzer, die diese auslösen, zusammen. Predict berechnet die Antwortzeit für jede Transaction im Workload. Weiter wird der Durchsatz berechnet, den ein Benutzer erzeugt.

(e) Disk

Disks repräsentieren physikalische Laufwerke und andere Laufwerke wie RAID Storage Arrays und CD-ROM und Tape-Drives auf jedem Knoten.

(f) Logical Volumes

Logical Volumes repräsentieren das Dateisystem, das sich über ein oder mehrere physikalische Laufwerke erstrecken kann. Alle betroffenen physikalischen Laufwerke müssen auf einem Knoten installiert sein.

04.04 Zusammenfassung

Als auf Agenten basierendes System ist BEST/1 in der Installation recht aufwändig. Um auf Datenbanken zugreifen zu können, muss eigens hierfür ein BEST/1-Datenbankbenutzer eingerichtet werden. Da auf jedem zu untersuchenden Rechner die Agenten installiert werden müssen, steigen die Kosten bei großen Netzwerken schnell an.

BEST/1 eignet sich sehr gut für die Gewinnung und Archivierung von Systemdaten in bestehenden Systemen. Die Darstellung von historischen Daten wird vom Visualizer recht gut unterstützt. Mit der Komponente Monitor lassen sich Systemdaten in Real-Zeit betrachten. Die Bestimmung von Antwortzeiten mit der Komponente Predict setzt genaue Kenntnis der Prozesse der betrachteten Transaktion voraus. Es müssen allerdings zusätzlich Daten eingegeben werden, die von BEST/1 nicht ermittelt werden.

Der Netzwerkverkehr wird in Predict auf der Grundlage des Netzwerk I/O-Verhaltens der Transaktionen berechnet. BEST/1 untersucht keine Datentransfer-Pakete.

KAPITEL 05 BEST/1 – EIN ERFAHRUNGSBERICHT

JÜRGEN LÜHRS

05.01 Einleitung

Es wird der Einsatz der BEST/1-Komponente Monitor in einem Kundenprojekt beschrieben. Die beschriebene IT-Umgebung und die geschilderten Probleme spiegeln die Realität wider, der Kunde bleibt jedoch anonym.

Für den Kunden wurde ein komplettes Systems Management eingerichtet. Es kam eine Vielzahl von Produkten zum Einsatz, wobei hier nur die Sachverhalte beschrieben werden, die für die Einrichtung von BEST/1 von Interesse sind. In einem ersten Schritt wurde die BEST/1-Komponente Monitor konfiguriert. In einer späteren Phase sollen die Tools zur BEST/1 Kapazitätsplanung (Manager, Collect, Analyze und Predict) eingesetzt werden.

05.02 Die IT-Umgebung

(a) Die Produktionsumgebung

Die IT-Umgebung des Kunden ist über viele Jahre gewachsen. Es finden sich Großrechner, UNIX-Systeme sowie Windows NT-Systeme in dieser Umgebung. Als Arbeitsplatzrechner wird Windows NT Workstation eingesetzt. Viele zentrale Anwendungen des Kunden werden heute noch auf Großrechnern gefahren, wie z.B. Warenwirtschaft und Job-Scheduling. Es ist geplant, im Laufe der nächsten Jahre die Großrechner auszumustern und die Anwendungen auf UNIX und NT zu portieren.

Für die Kapazitätsplanung und das Performance Management mit BEST/1 werden die UNIX- und NT-Server betrachtet.

@ UNIX: Bei den UNIX-Systemen handelt es sich um IBM AIX Systeme. Sie kommen als allein stehende Maschinen oder als Cluster vor. Die Cluster bestehen aus 2 Maschinen und werden über die Hochverfügbarkeitslösung HACMP von IBM verwaltet. Die UNIX-Maschinen dienen als Datenbankserver und als Server mit firmenspezifischer Software.

- Windows NT: Die Windows NT Server kommen als alleinstehende Server und als Windows NT Cluster (2 NT Maschinen) vor. Sie dienen als File-Server, Fax-Server und als Server mit firmenspezifischer Software.
- Datenbanken: Die Produktionsdaten werden in Oracle Datenbanken gehalten. Die Oracle Datenbanken kommen auf den allein stehenden UNIX-Servern und den UNIX-Clustern vor. Auf einem UNIX-Server können mehrere Datenbanken vorhanden sein.
- Der Sitz des Kunden verteilt sich auf mehrere Standorte. Es gibt eine Hauptverwaltung und mehrere Niederlassungen. Die Anbindung der Niederlassungen erfolgt über ISDN. Das verwendete Protokoll ist TCP/IP. Die Anbindung der Großrechner erfolgt über einen Token Ring.

05.03 Die Systems Management Knoten

Für die Systems Management Tools wurde ein eigener HACMP-Cluster verwendet. Er besteht aus zwei AIX Maschinen. BEST/1 wurde auf einem Shared Volume installiert, so dass bei einem Ausfall einer Cluster Maschine die Applikation von der anderen Maschine übernommen wird. So wird eine hohe Verfügbarkeit des BEST/1 Monitors gewährleistet. Die Performance der Systems Management Knoten wird ebenfalls von BEST/1 überwacht.

05.04 Anforderungen des Kunden

(a) Allgemeine Performanceüberwachung

Der Kunde forderte von BEST/1 eine Überwachung der allgemeinen System- und Datenbankparameter. Die wichtigsten zu überwachenden Objekte für die Betriebssysteme sind hierbei

- CPU, Memory, Swap, Prozesse
- Filesystem, Disks
- sowie für die Oracle Datenbanken User Sessions

Für diese Objekte werden eine Vielzahl von Metrics (Parameter) erhoben, die zur späteren Analyse in einem History Repository abgespeichert werden. Eine detaillierte Aufstellung der Vorstellungen des Kunden über die Anforderungen an die Kapazitätsplanung und das Performance Management sind im Abschnitt 05.07 ‚Anhang' aufgeführt.

(b) Spezielle Performanceprobleme

Neben der allgemeinen Überwachung der Performancedaten hat der Kunde ein spezielles Performanceproblem. Die Antwortzeiten beim Zugriff auf die Datenbanken sind in der Regel zufriedenstellend. Manchmal verschlechtern sich die Antwortzeiten auf Werte bis zu mehreren Minuten. Das ist nicht mehr akzeptabel. Die betroffenen Mitarbeiter beschweren sich und die Produktivität sinkt. Dem Kunden ist es bisher nicht gelungen, die Ursache für die temporäre lange Antwortzeit festzustellen. Als Instrument zur Analyse stand bisher nur ein Paket-Sniffer zur Verfügung, mit dessen Hilfe der Datenverkehr zwischen dem anfragenden Client und dem Datenbankserver untersucht wurde. Da nur erst nach einer telefonischen Beschwerde gehandelt werden konnte, war das Performanceproblem bisher immer schon verschwunden, bevor aussagekräftige Snifferdaten erhoben werden konnten. Hier erhofft sich der Kunde von BEST/1 Hinweise zur Problemlösung. Es wird vermutet, dass die Ursache des Problems kein Netzwerkproblem ist, sondern auf dem Datenbankserver zu finden ist.

(c) Lösungen

Der BEST/1 Monitor wird zur Unterstützung bei der Fehleranalyse auf der Basis von Realzeitbetrachtungen und der Analyse von historischen Daten eingesetzt. Von der Möglichkeit, Schwellwerte anzugeben und bei deren Überschreitung ein Alarm Event zu generieren, wurde zu Gunsten eines anderen BMC Produkts, BMC Patrol, in diesem Projekt kein Gebrauch gemacht. Im Vordergrund stand die Unterstützung bei der Suche nach den Ursachen von Performanceengpässen in den Systemen und Datenbanken. Die Grundidee besteht darin, nach dem Auftreten eines Performanceproblems die Möglichkeit zu haben, dieses detailliert anhand der historischen Daten untersuchen zu können. Der BEST/1 Monitor bietet die Möglichkeit, zu einem beliebigen Zeitpunkt aus den historischen Daten verschiedene Metrics darzustellen. So kann z.B. bei einer hohen Auslastung der Datenbank die CPU-Zeit jedes einzelnen Datenbankbenutzers ermittelt und daraus auf einen möglichen Verursacher eines temporären Performanceproblems geschlossen werden. Voraussetzung hierfür ist eine feine Granularität der historischen Daten. Die Default Einstellung des BEST/1 Monitors stellt folgende Auflösung zur Verfügung:

- @ 1 Stunde in 10 Sekunden Intervallen
- @ 8 Stunden in 5 Minuten Intervallen
- @ 40 Stunden in 15 Minuten Intervallen

Dies ergibt eine Aufzeichnungsdauer von ca. 2 Tagen. Der hierfür benötigte Speicherplatz wird mit ca. 750 MB angegeben. Die Aufzeichnungsdauer und die angebotene Auflösung erschien dem Kunden für die oben genannten Anwendungen zu wenig. Die Einstellungen

sollen auf eine Aufzeichnungsdauer von 5 Tagen in 10 Sekunden Intervallen geändert werden. Die Größe des benötigten Speicherplatzes wird auf ca. 5 Gigabyte geschätzt, ein recht beachtlicher Wert, zumal dieser Speicherplatz für jeden zu überwachenden Rechner bereitzustellen ist.

Zur Messung von Antwortzeiten bietet der BEST/1 Monitor in der zum Einsatz kommenden Version 6.2 den Standard ARM (Application Response Measurement) an. Dieses Feature wurde in diesem Projekt (noch) nicht eingesetzt.

05.05 Die Realisierung

(a) Installation

Der BEST/1 Monitor mit seiner GUI zur Konfiguration und Anzeige der Messergebnisse wurde auf dem Systems Management Knoten installiert. Auf allen zu überwachenden Rechnern (UNIX und NT) wurden der BEST/1 Agent und die zugehörenden Kollektoren installiert. Die Installation erfolgte problemlos.

Bei der Inbetriebnahme der Installation kam es aber zu folgenden Problemen:

Der Versuch, von einem zu überwachenden Knoten die Performancedaten zu erheben, schlug fehl. Es zeigte sich, dass der BEST/1 Monitor die Agenten auf den zu überwachenden Knoten nicht kontaktieren konnte. Dieser Tatbestand wurde an den Support des Herstellers (BMC) weitergeleitet. Der Hersteller lieferte daraufhin 2 Patches. Der BEST/1 Agent (bgsagent) und der Service-Daemon (bgssd) mussten ausgetauscht werden. Danach konnten Performancedaten erhoben werden. Die hiermit verbundenen Verzögerungen im Projekt betrugen mehrere Tage.

Im laufenden Betrieb in der Testumgebung wurde ein großer Speicherplatzbedarf des UNIX System-Kollektors (bgscollect) festgestellt. Der Speicherplatzbedarf dieses Prozesses wuchs auf 450 MB, bis der Kunde die Abschaltung forderte. Der Support des Herstellers lieferte daraufhin einen weiteren Patch. Der UNIX System-Kollektor wurde ausgetauscht. Das Problem schien damit gelöst. Eine weitere Projektverzögerung war aber die Folge.

Nach längerer Zeit wuchs auch der Speicherplatzbedarf dieses Patches auf mehrere 100 MB. Der Hersteller lieferte einen weiteren Patch. Wiederum wurde der UNIX System-Kollektor ausgetauscht. Der Ergebnis dieser Aktion stand zum Zeitpunkt der Niederschrift noch nicht fest.

Zur Abnahme des Testfeldes wurde im Rahmen von „Testcases“ einige der von BEST/1 Monitor ermittelten Performancedaten einer Überprüfung unterzogen. Es zeigte sich, dass einige Werte von BEST/1 Monitor falsch ermittelt werden. Dies bezieht sich auf Daten, die

von den System-Kollektoren ermittelt werden, sowie auf Daten die von den Oracle-Kollektoren ermittelt werden. Der Support des Herstellers lieferte daraufhin wieder einen Patch. Der Oracle-Kollector (bgsoracollect.exe) wurde ausgetauscht. Dieser Patch konnte allerdings nicht alle Probleme beheben. Das Projekt wurde wiederum verzögert.

Zusammenfassend lässt sich sagen, dass die Installation und die Inbetriebnahme geprägt waren durch viele Produktfehler und durch lange Antwortzeiten des Supports des Herstellers.

05.06 Konfiguration

Die Konfiguration der zu überwachenden Rechner ist recht einfach. Mit Hilfe eines Discovery-Mechanismus lassen sich die mit BEST/1 Agenten installierten Rechner im Netz entdecken und auf der GUI des Monitors als Objekt darstellen. Die Anbindung der Oracle Datenbanken stellte ebenfalls kein Problem dar. Hierzu muss auf den zu überwachenden Datenbankservern in jeder Datenbankinstanz ein SQL-Skript laufen, das einen Datenbankbenutzer anlegt und diesem die benötigten Rechte auf die Tablespaces der Performanceparameter gewährt. Die Datenbank lässt sich mit einem Subsystem-Objekt auf der Monitor-GUI darstellen.

Problematischer erwies sich das Einrichten eines Mechanismus zum automatischen Starten und Stoppen der BEST/1 Prozesse. Die Agenten und Kollektoren sollen nach einem Neustart der Maschine automatisch gestartet werden, um eine lückenlose Aufzeichnung der History-Daten zu gewährleisten. Das mitgelieferte Startskript startet nur den Agenten und den Kollektor für Betriebssystem-Daten, nicht aber den Kollektor für Oracle-Daten. Der automatische Start des Oracle-Kollektors ist nur über eine Policy möglich, die in der Monitor-GUI konfiguriert wird.

Die Einstellung der History auf eine Aufzeichnungsdauer von 5 Tagen in 10 Sekunden Intervallen erwies sich ebenfalls als problematisch. Für diese Einstellung ist kein Menüpunkt in der GUI vorgesehen. Es musste eine Konfigurationsdatei angepasst werden. Ein Test der History-Daten nach einigen Tagen zeigte, dass die History-Daten nicht gemäß den Einstellungen in der Konfigurationsdatei erhoben werden. Bei dieser Größe der History steigt die Zugriffszeit auf die Daten merklich an.

Ein weiteres Problem stellt die Verwendung von BEST/1 in Hochverfügbarkeitsumgebungen wie HACMP-Cluster oder Windows NT-Cluster dar. Hier muss für den zu überwachenden Rechner eine IP-Adresse angegeben werden, die unabhängig vom Zustand der Hochverfügbarkeitslösung stets auf denselben Rechner zeigt.

(a) Betrieb

Der BEST/1 Monitor ermittelt eine Vielzahl an Performancewerten. Die geforderten Standardparameter können gut beobachtet werden. Die Bedienung ist recht einfach und schnell zu erlernen.

Werden viele Objekte in der GUI angelegt, wie es in der Regel der Fall ist, wirkt die Darstellung schnell unübersichtlich. Die Bedeutung der angezeigten Metrics muss genau studiert werden. Dies setzt die Lektüre des zugehörigen Handbuchs voraus. Die Erläuterungen der Metrics in diesem Handbuch fallen manchmal sehr dürftig aus. Dies verschlechtert die Akzeptanz des Produktes beim Anwender.

Der Kunde wünschte sich nicht nur die Präsentation von Daten, sondern auch eine Unterstützung bei deren Interpretation. Dies kann ein Produkt wie BEST/1 Monitor allerdings nicht bieten. Für Verdruss sorgten die vielen Produktfehler und die langen Reaktionszeiten des Hersteller-Supports.

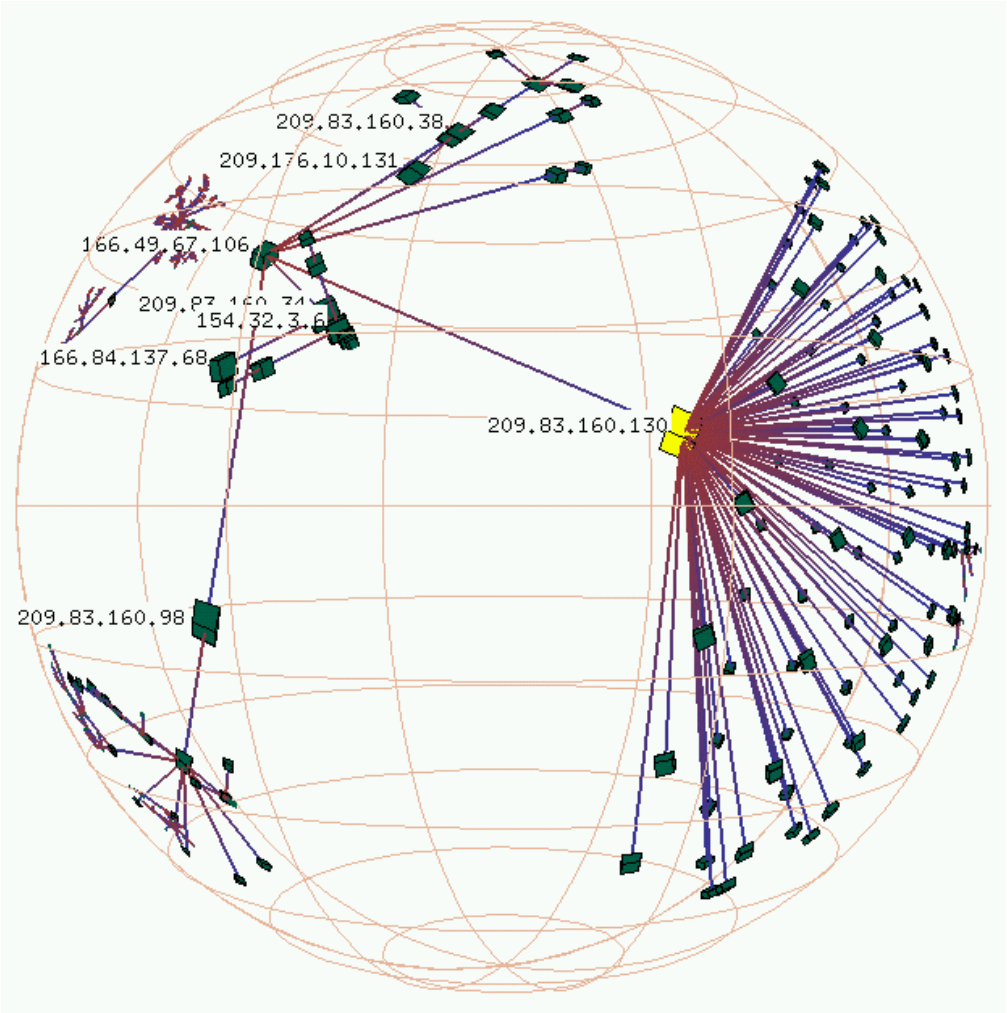
05.07 Die Anforderungen in Stichworten

- CPU Hauptspeicherauslastung, Plattenbelegung
- Kombination der Werte einstellbar
- auf Prozessor bezogene Werte
- Kapazitätsüberwachung User-bezogen, Anwender-bezogen, jeweils für Platte, CPU, sonst.
- Performanceüberwachung: Anwendung, User, Historie
- Trendanalysen und Prognosen, generell (CPU, I/O, Paging, Netz)
- Spoolinganforderungen/-Einträge
- Unterstützung bei Fehleranalyse Historiendaten
- automatische Teilbereinigung nach Schwellwerten bzw. Kriterien
- Trace Aufzeichnung und Auswertung, zentrales Systemprotokoll pro Maschine Auswertung mit Selektion und Filterung
- Feststellen von Verursachern bei Performanceproblemen und in diesem Fall mind.: Anzeige und Analyse der laufenden Prozesse
- (z. B. Anzahl und Anteil CPU) sowie eine User-bezogene Anzeige/Analyse
- Performance Speicher/Paging: I/O p. sec., Auslastung File-System, MB p. sec., SCSI Controller Verhalten etc.

- @ Zusammenstellung von Performance Kennzahlen anwendungsbezogen
- @ Historiendaten sind möglich
- @ Überwachung mehrerer dezentraler Systeme, Überwachung und Verwaltung von Ausnahmesituationen im Netzwerk
- @ Ermittlung und Lösung von Problemsituationen
- @ Überwachung von: Systemverfügbarkeit, Dateiauslastung, Auslastung des Swapspace, Prozess-Status, Vorhandensein von Objekten, Kommunikationsverbindungen
- @ Weiterleitung eines Performanceproblems an die für die Problemlösung zuständige Person
- @ RMON Unterstützung
- @ Auslastungsanalysen: aktuelle Be-/Auslastung einer Ressource. Wie hat sich die durchschnittliche Auslastung entwickelt?
- @ Trend, zu erwartende Auslastung
- @ Kombination in Verbindung mit Ressourcen eines Prozesses (Prozessverfolgung)
- @ Langzeitauswertungen
- @ Antwortzeitmessung, Punkt zu Punkt Messung, auch segmentübergreifend
- @ Graphische Anzeige der Systemleistung, Echtzeitaufzeichnung von Speicherauslastung, CPU-Auslastung und CPU-Eigenschaften
- @ Echzeit-Aufzeichnung des Prozessstatus: Wer ist „owner“ des Prozesses und welche Ressourcen werden genutzt?
- @ Echtzeitaufzeichnung des Gerätestatus: Wie hoch ist die Aktivität, wie performant wird gearbeitet?
- @ Oracle Ressourcenverbrauch/Schwellwertüberschreitung: Antwortzeiten (im Durchschnitt); REDO-Logfile (in %); Switchhäufigkeit als absoluter Wert; Plattenkapazität (in %); CPU-Anteil (in %), I/O Verhalten bezogen auf Platte bzw. File
- @ Oracle Problemanalyse: a) User-bezogen bzw. b) Anwendungsprozess-bezogen bzw. c) Oracleprozess-bezogen: mit CPU, I/O, Tabellen, Lock, GSA, Hauptspeicher, Unix-Shadow Prozesse
- @ Oracle Historiendaten mit folgenden Werten: CPU-Anteil, I/O Verhalten Switchhäufigkeit – Antwortzeiten, Tabellen-Zugriffe, I/O Verhalten Volume/Platte
- @ Oracle Historiendaten mit Anteil an den Gesamtressourcen: CPU-Anteil, I/O Verhalten, Antwortzeiten
- @ Oracle Überwachung der alert-files (inhaltlich)

- Oracle Überwachung der Betriebs-Prozesse
- Integration zu anderen Komponenten und Funktionen des Service Managements
- Überwachung Replikation: Größe der Log-Tabellen, Dauer der Pull/Push-Prozesse

Teil V: Web und Intranet



Inhalt Teil V

KAPITEL 01 DAS WERKZEUG WEBLOAD 3

01.01 ÜBERBLICK 3
01.02 ANFORDERUNGEN 4
01.03 EVALUIERUNG 5
01.04 FAZIT 7

KAPITEL 02 UNTERSUCHUNG DES WEB-ACCOUNTING 9

02.01 EINLEITUNG 9
02.02 WELCHE LOG-DATEIEN GIBT ES? 9
02.03 WELCHE INFORMATIONEN LASSEN SICH GEWINNEN? 10
02.04 WELCHE INFORMATIONEN LASSEN SICH NICHT GEWINNEN? 10
02.05 UNBEGRÜNDETE SCHLUSSFOLGERUNGEN 11
02.06 WAS IST BEI SCHLUSSFOLGERUNGEN AUßERDEM ZU BEACHTEN? 11
02.07 BEISPIEL 11

KAPITEL 03 CLIENT/SERVER PERFORMANCE-ANALYSE 12

03.01 EINLEITUNG 12
03.02 ANFORDERUNGEN AN MESSDATEN 13
03.03 TRANSAKTIONSVERFOLGUNG MIT (WEB)TRACY 16
03.04 FALLSTUDIE 20
03.05 ERGÄNZENDE LITERATUR 27

KAPITEL 01 DAS WERKZEUG WEBLOAD

NORBERT SCHERNER

01.01 Überblick

WebLOAD ist ein Tool der Firma Radview zum Test von e-Commerce und e-Business Web-Applikationen. Es integriert dabei den Leistungsumfang eines Lasttreibers zu allen anderen Mitteln für den Funktionalitätstest. Bei der nachfolgenden Bewertung werden jedoch ausschließlich die Aspekte von WebLOAD betrachtet, die den Einsatz als Lasttreiber betreffen. Die Funktionalität als Test-Tool findet keine Berücksichtigung.

WebLOAD basiert nicht auf einer Agenten-Technik mit UNIX- oder NT-Agenten. Die Erzeugung einer Last in Form von HTTP-Requests erfolgt durch einen oder mehrere Treiber-Rechner, die durch eine sogenannte Konsole gesteuert werden. Im einfachsten Fall sind Konsole und Treiber-Rechner auf einer Maschine.

Mittels einer Recording-Facility können HTTP-Requests einer realen Browser-Sitzung aufgezeichnet und in Skripten – in WebLOAD Agenden genannt – abgelegt werden. Die Skripten basieren auf der Sprache JavaScript und unterstützen Standards wie DOM, XML, SSL und Proxies. Es ist aber auch eine künstliche Erzeugung einer Agenda mittels eines speziellen Editors möglich. Die Skripte können weiter editiert und mit Kontroll-Strukturen sowie Variablen versehen werden.

Nach einem Syntax-Check sind sie auf den Treiber-Rechnern ablauffähig und emulieren dann einen oder mehrere Clients. Während des Playbacks der Agenden werden umfangreiche Antwortzeit-, Last- und Test-Statistiken gesammelt. Diese werden online auf der Konsole angezeigt und in einer MS-Access-Datenbank abgelegt.

Die zu anderen Produkten vergleichbare Funktionalität wurde um Features ergänzt, welche bei manch anderem Produkt nicht verfügbar sind. Dazu gehören Simulation unterschiedlicher Bandbreiten, Browser-Caching, benutzerdefinierte Lastziele, Konfigurations-Wizards, Datenanalyse über unterschiedliche Ebenen, individuell anpassbare Zeitstrecken und Zähler sowie umfangreiche Performance-Überwachung der am Test beteiligten Server.

01.02 Anforderungen

Bei der Bewertung von Lasttreibern sind als State-of-the-Art heutzutage folgende Punkte von Bedeutung:

(a) Skripts

- Aufzeichnung von Ready-to-Use Skripten
- Editierbarkeit der Skripten (Variablen, Kontroll-Strukturen, Timer)
- Einfache Skript-Sprache
- Automatische Generierung von Test-Daten-Namen/Feldern
- Umwandlung von Log-Dateien in Skripts

(b) Playback

- Single/Multi-User Playback
- Synchronisation innerhalb des Skripts und zu anderen Skripten
- Verwendung von variablen Input-Daten
- Zeitsteuerung beim Playback
- Stabilität und Verhalten bei Abweichungen beim Playback

(c) Logging

- Logging von Aufzeichnung/Playback
- Einfache Analyse der Logging Informationen

(d) Statistiken

- Konfigurierbare Statistiken
- Import der Statistiken

(e) Überwachung

- Graphische Aufbereitung beim Playback
- Parallele Überwachung aller beteiligten Systeme
- Dynamische Anpassung der Metriken

01.03 Evaluierung

WebLOAD besteht aus einem Installations-Paket für NT- und 2000-Plattformen. Zusätzlich können mit dem WebLOAD-Resource-Manager weitere Rechner – auch Sun-Solaris-Rechner – als Treiber verwaltet werden. Auf der Konsole werden Abläufe erzeugt, gestartet, beobachtet und Ergebnisse analysiert. Auf dem/den Treiber-Rechner(n) (NT, 2000 oder UNIX) laufen die gestarteten Skripte.

(a) Skripts

Es gibt zwei Wege ein Skript bzw. eine Agenda zu erzeugen. Ein Weg führt über das sogenannte Agenda Authoring Tool (AAT) zur Aufzeichnung der Eingaben in einem Web-Browser. Während der Aufzeichnung können durch den Wechsel zwischen Browser und AAT Kommentare, Timer, Transaktionsbegrenzer, Funktionsblöcke und weitere Elemente der JavaScript-Sprache in die Aufzeichnung übernommen werden. Nach dem Beenden der Aufzeichnung werden das aufgezeichnete Skript sowie Dateien, welche Eingaben zu Query-Aufrufen als externe Datenquellen für das Playback enthalten, abgelegt.

Die entstandenen Agenden sind nach korrekt durchlaufenem Syntax-Check auf einem Treiber-Rechner ablauffähig, um die aufgezeichneten Aktivitäten des Client zu wiederholen.

Neben der Aufzeichnung einer Browser-Session kann sowohl über den in WebLOAD integrierten als auch über jeden beliebigen Standard-Editor in Anlehnung an die JavaScript-Sprache eine Agenda erstellt werden. Über Konstrukte der Sprache Java sind weitere Möglichkeiten der Editierung der Skripts vorhanden. Damit können auch allgemeine Aufrufe zur Steuerung des Benutzerverhaltens, zusätzliche Transaktions-Strecken, Konsolen-Kommunikation und Ein/Ausgabe aus/in Dateien ergänzt werden.

Die Möglichkeit, Web-Server-Log-Dateien in Agenden umzuwandeln besteht nicht.

Ob der Ablauf des Programms als Playback funktioniert, ist generell von zwei Faktoren abhängig:

- Ist der Vorgang an sich wiederholbar? I.d.R. sind dies lesende Transaktionen, während z.B. das Löschen ein und desselben Datensatzes in einer Datenbank nicht wiederholbar ist.
- Vergibt der Server von der jeweiligen Session abhängige ID's, die Bestandteil der weiteren Kommunikation sind? Ist dies der Fall, so ist selbst das Lesen von Daten nicht ohne eine Modifikation des Skripts wiederholbar.

Hier stellt WebLOAD diverse Funktionsaufrufe zur Verfügung, welche die oben genannten Probleme lösen.

(b) Playback

Fertige und übersetzte Skripts, d.h. Skripts mit den notwendigen Modifikationen um einen wiederholbaren Ablauf sicherzustellen, können beliebig oft und in beliebiger Anzahl parallel (genügend Rechnerkapazität des Treiber-Rechners vorausgesetzt) zum Ablauf gebracht werden. Der gleichzeitige Start mehrerer Skripts wird dabei über die Konsole bewerkstelligt. Dabei ist es auch möglich, Skripts auf mehreren Treiber-Rechnern zu starten, um ggf. die notwendige Rechnerkapazität zur Verfügung zu haben. Der Ablauf beim Playback (Anzahl Clients, Dauer, Zielauslastung u.a.m.) wird in konfigurierbaren Templates beschrieben und gespeichert.

Ein Aspekt beim Playback ist die Synchronisation. Innerhalb eines Skripts versteht man darunter die Eigenschaft, dass der nächste Aufruf an den Server erst dann erfolgt, wenn eine Antwort zum vorausgegangenen Aufruf vollständig zurückgekommen ist oder negativ quittiert wurde. WebLOAD nutzt hier die Funktionalität des HTTP-Protokolls, welches erfolglose Requests durch einen Timeout beendet. Abgesehen davon, dass Zeitüberschreitungen bei den Antworten sowie Fehler beim HTTP-Protokoll abgefangen werden, sind sonstige Abweichungen beim Playback im Skript selbst zu programmieren. Damit kann auf Fehler mit dem Abbruch des Skripts reagiert werden. Aber auch zwischen Skripts kann untereinander eine Synchronisation erforderlich sein. Hier bietet WebLOAD die Möglichkeit, die Synchronisation über einen speziellen Aufruf der Skript-Sprache zu realisieren.

Unter der Verwendung von variablen Input-Daten versteht man, dass sichergestellt ist, dass gleiche Skripts mit unterschiedlichen Input-Daten arbeiten können. Dies erfolgt bei WebLOAD durch Ersetzen der variablen Daten zu einem Skript aus dem Wertevorrat von Input-Dateien, die sequentiell oder zufallsgesteuert gelesen werden.

Im Multi-User-Playback können mehrere Agenden zu einem Mix kombiniert werden.

Neben der Zeitsteuerung in einem Skript kann als zeitliche Begrenzung des Ablaufs eine Laufzeit oder das Erreichen eines Lastzieles definiert werden.

(c) Logging

Beim Playback werden Log-Informationen, die sämtliche Status-Ausgaben vom Web-Server enthalten, erstellt und in einer MS-Access-Datenbank abgelegt. In der Log-Information des Playbacks sind auch die Zeitstempel abgelegt, die online auf dem Konsole ausgewertet werden können. Durch ein sogenanntes Data-Drilling kann die Ebene der Analyse verfeinert werden.

(d) Statistiken

Wie schon festgehalten, werden die Informationen beim Playback in eine Datenbank eingetragen. Die extrahierten Informationen können dann in eine Statistik umgewandelt werden. Diese Statistik kann entweder im Tab-Format oder nach MS-Excel importiert werden. Zu den Konfigurationsmöglichkeiten der Reports gehören neben einer Reihe von Standards (Load Size, Transaktionen pro Sekunde, Durchsatz in Byte pro Sekunde) auch die Zähler des Performance-Monitors PERFMON sowie RSTATD unter UNIX.

Da die Konsole über eine graphische Oberfläche verfügt, gestaltet sich auch die online-Auswertung der Statistiken unter WebLOAD sehr komfortabel. Über das Data-Drilling können nicht nur Zeitstempel sondern auch Informationen zum Ablauf (HTTP-Statistiken, Datenübertragungsvolumen, Fehler) ausgewertet werden, sofern die entsprechenden Zähler bzw. Events bei der Konfigurierung einer Load-Session aktiviert wurden.

(e) Überwachung

Beim Playback ist auf der Konsole eine online-Ablaufüberwachung der Skripten möglich. Wesentliche Informationen zum Skript-Ablauf werden tabellarisch oder als Histogramm auf dem Bildschirm angezeigt. Es ist auch möglich, Informationen aus dem Skript (z.B. Schleifenzähler) zur Ausgabe auf die Konsole umzuleiten.

01.04 Fazit

WebLOAD ist für Lasttests und Benchmarks im Bereich von Web-Applikationen sehr gut einsetzbar. Kenntnisse der Sprache Java vorausgesetzt, ist die Skript-Sprache leicht zu erlernen und das Tool einfach zu bedienen. Durch die Vereinheitlichung der Tool-Landschaft (Funktionstest und Performancetest) ist eine bessere Nutzung vorhandener Ressourcen möglich. Über die Aufteilung in Konsole und Treiber-Rechner wird eine gute Skalierbarkeit sichergestellt. Es fehlen nur wenige Features wie die Umwandlung von Web-Server-Log-Dateien, automatische Checkpoints und die Sammlung von Statistiken aus Datenbanken.

(a) Ausblick

Die Evaluierung wurde mit WebLOAD 4.0 durchgeführt. Die Version 4.5 (ab Ende 2000) besitzt zusätzlich folgende neue Features:

- HTML-gestützte Reports
- SNMP-Interface zur Einbeziehung von Metriken diverser third-party-Plattformen wie Applikation-Server und Datenbank-Server

- @ Wizard für automatische Checkpoints und
- @ Visueller step-by-step-Durchlauf von Agenden

KAPITEL 02 UNTERSUCHUNG DES WEB-ACCOUNTING

NORBERT SCHERNER

02.01 Einleitung

Je mehr das World-Wide-Web wesentlicher Bestandteil des Geschäfts und der Geschäftsbeziehungen von Firmen wird, um so mehr wächst der Bedarf an Informationen über die Performance der Server, die im Dienst des Web stehen. Aus diesem Grund gewinnt die Beobachtung und die Planung der Web-Server-Kapazitäten immer mehr an Bedeutung. Die Nutzung von Web-Server-Log-Dateien bietet hier eine Basis für Statistiken über Nutzungsgrad und Wachstum einer Website über definierte Zeiträume. Die Analyse dieser Daten gibt außerdem Aufschluss über die Belastung des Web-Servers, ungewöhnliche Aktivitäten oder erfolglose Anfragen. Damit kann sie sowohl dem Marketing als auch dem IT-Management nützliche Angaben zur kontinuierlichen Leistungskontrolle geben.

02.02 Welche Log-Dateien gibt es?

Jede Kommunikation zwischen einem Browser auf einem Web-Client und dem Web-Server resultiert in einem Eintrag in der Log-Datei des Web-Servers. Die Daten, die in der Log-Datei gesammelt werden, variieren je nach Typ und Betriebssystem des Web-Servers sowie dem Format, das dieser unterstützt. Am weitesten verbreitet sind das „common log file format“ und das „combined or extended log file format“.

Im allgemeinen enthält jeder Log-Datei-Datensatz folgende Informationen:

- die Adresse des Rechners, der die Seite angefordert hat
- Datum und Uhrzeit der Anforderung
- die URL der Seitenanforderung
- die Größe der angeforderten Seite

Je nach Typ können noch folgende weitere Informationen gespeichert sein:

- der Browser und das Betriebssystem des anfordernden Rechners
- das für die Anforderung benutzte Protokoll

- Zeit für die Verarbeitung der Anforderung
- HTTP-Status der Anforderung
- die URL der Seite, dessen Referenz angefordert wurde

Die wichtigsten Log-Formate sind die von Netscape, MS, Website, NCSA und WebSTAR.

02.03 Welche Informationen lassen sich gewinnen?

Die Daten der Log-Dateien lassen sich auf vielfältige Weise verdichten, um folgende Informationen zu gewinnen:

- Anzahl der Anfragen („hits“)
- Anzahl der Seiten und Kilobytes erfolgreich geliefert
- welche IP-Adressen wurden bedient mit Zuordnung der Anzahl Anfragen
- Anzahl der Anfragen verteilt nach HTTP-Status (erfolgreich, erfolglos, weitergeleitet)
- Summe und Durchschnitt über definierte Zeiträume (Stunden, Tage, Wochen, Monate, Jahre)
- Anzahl der Anfragen auf bestimmte Seiten oder Directories
- Anzahl der Anfragen verteilt auf Domänen (abgeleitet von IP-Adressen)

Diese können dann in Tagesübersichten, Wochen-, Monats- bzw. Jahresberichten oder Domänen-Statistiken zusammengefasst werden.

02.04 Welche Informationen lassen sich nicht gewinnen?

Die Unzulänglichkeiten der Log-Daten lassen sich in drei Bereiche einordnen:

- Datentyp wird nicht gelogged
- Log-Daten unvollständig
- unbegründete Schlussfolgerungen aus den Log-Daten

Daten, die nicht gelogged werden sind z.B.

- Anzahl Benutzer, da sich aus der geloggten IP-Adresse keine 1-zu-1-Beziehung zu einer Person herleiten lässt (Beispiel: Internet-Service-Provider)
- individuelle Daten wie Namen oder e-Mail-Adressen
- Trace-Daten wie die Aufeinanderfolge besuchter Seiten

@ Unvollständige Daten

Die Anzahl der Anfragen ist abhängig von den Cache-Einstellungen auf dem Weg vom Server zum Client.

02.05 Unbegründete Schlussfolgerungen

Eine Reihe von Schlussflgerungen aus den Log-Daten sind entweder unzulässig oder nicht aus-reichend gerechtfertigt. Dazu gehören:

@ „hits“ entspricht Nutzung

@ Benutzer-Sessions aus gleichen IP-Adressen

@ Beliebtheit bestimmter Seiten.

02.06 Was ist bei Schlussfolgerungen außerdem zu beachten?

Die statistische Analyse des Web-Accounting erfordert gewisse Annahmen bzgl. der Daten. So sollten die Daten annähernd normal verteilt sein. Aus den bisher gemachten Erfahrungen lässt sich jedoch festhalten, dass Kurzzeit-Daten extrem schwanken können. Nur durch die Beobachtung über mehrere Monate ist es möglich, zwischen Trend und Schwankung zu unterscheiden. Wegen der hohen Schwankungsbreite ist es durchaus angeraten, genug freie Server-Kapazität zu haben, um etwa die doppelte Last wie im Mittel ohne Probleme bewältigen zu können.

02.07 Beispiel

requesting host		Datum	Uhrzeit			WEB-Server			transferd Bytes	HTTP Status			requested file	
193.98.99.154	-	18.04.00	00:03:17	W3SVC	PROFI1	139.25.96.62	0	490	121	404	3	GET	/ralph/131.htm	-
193.98.99.154	-	18.04.00	00:03:44	W3SVC	PROFI1	139.25.96.62	63	501	3959	200	0	GET	/Inf-sek2/home/o-euro.htm	-
193.98.99.154	-	18.04.00	00:04:01	W3SVC	PROFI1	139.25.96.62	125	498	16999	200	0	GET	/SEA/Zeit-ist-reif.htm	-
193.98.99.154	-	18.04.00	00:04:07	W3SVC	PROFI1	139.25.96.62	125	502	7271	200	0	GET	/Inf-Sek2/euro/drucker.htm	-
193.98.99.154	-	18.04.00	00:04:11	W3SVC	PROFI1	139.25.96.62	63	501	3747	200	0	GET	/Inf-Sek2/handy/index.htm	-
193.98.99.154	-	18.04.00	00:04:20	W3SVC	PROFI1	139.25.96.62	63	503	3959	200	0	GET	/Inf-sek2/home/o-websit.htm	-
193.98.99.154	-	18.04.00	00:04:26	W3SVC	PROFI1	139.25.96.62	78	503	6049	200	0	GET	/Inf-Sek2/email/webmail.htm	-
193.98.99.154	-	18.04.00	00:04:30	W3SVC	PROFI1	139.25.96.62	63	502	3959	200	0	GET	/Inf-sek2/home/o-email.htm	-

Tabelle 02-1: MS IIS logfile format

KAPITEL 03 CLIENT/SERVER PERFORMANCE-ANALYSE

CORINNA FLÜS

03.01 Einleitung

Die Leistungsfähigkeit und Verlässlichkeit der zunehmend komplexer werdenden informationstechnischen Systeme sind für alle Organisationen von außerordentlicher Bedeutung. Insbesondere das exponentielle Wachstum des Internets und seines Benutzerkreises erfordert eine gewissenhafte Planung und Bereitstellung der benötigten Ressourcen, da die Anwender längeren Antwortzeiten extrem kritisch gegenüberstehen. Web-Seiten werden in der Regel nach ca. 8 Sekunden verlassen, wenn die gewünschten Daten nicht angezeigt werden. Da zunehmend auch Unternehmensumsätze über Webseiten erzielt werden, kann ein frühzeitiges Aussteigen der Anwender deutliche Umsatzeinbußen zur Folge haben. Anforderungen und Ziele von IT-Anwendern sollten daher den Ausgangspunkt für Performance-Analysen bilden.

Die Einhaltung von Dienstgüteanforderungen (*Service Level Agreements, SLA*s) der IT-Anwender spielt hierbei eine zentrale Rolle. Hierzu ist es erforderlich, das IT-System zu überwachen (zu vermessen), die Einhaltung oder Verletzung der SLAs festzustellen und gegebenenfalls Tuningmaßnahmen vorzunehmen. Die Gewinnung von aussagekräftigen Messdaten, die eine Überprüfung der Einhaltung der SLAs und eine Systemanalyse ermöglichen, stellt in der Praxis bereits eine große Herausforderung dar.

Dieses Kapitel beschreibt Lösungswege zur Gewinnung aussagekräftiger Performance-Daten für Client/Server-Anwendungen, welche insbesondere auch eine Überwachung der Einhaltung von SLAs ermöglichen. Die Performance-Daten werden hierbei aus Ereignis-Traces gewonnen, die aus Messungen in einem Client/Server-System hervorgehen. Zunächst werden die Anforderungen, die an Messdaten für eine Dienstgüteüberwachung gestellt werden und Lösungswege beschrieben. Abschließend wird eine Fallstudie aus dem Internet-Umfeld präsentiert.

03.02 Anforderungen an Messdaten

(a) Anforderungen für eine Dienstgüteüberwachung

Eine häufig anzutreffende Dienstgüteanforderung besteht in der Einhaltung bestimmter oberer Grenzen für Antwortzeiten. Diesbezügliche Service-Level-Agreements zwischen Dienstleistern und Kunden werden häufig nicht in Form von Mittelwerten angegeben, sondern vielmehr in Form detaillierter quantitativer Aussagen wie „die Antwortzeit einer Anwendung soll bei 90 % der Transaktionen 2 Sekunden nicht übersteigen“. Hierfür ist eine Mittelwertbetrachtung nicht ausreichend, es müssen Antwortzeitquantile untersucht werden.

Quantile werden folgendermaßen definiert:

Das *p-Quantil* bildet quasi - exakt nur im stetigen Fall - die Umkehrfunktion zur Verteilungsfunktion. Es ist nach [1] wie folgt definiert:

Seien $x_1, ..., x_n$ eine geordnete Messreihe und p eine reelle Zahl mit $0 < p < 1$, so bezeichnet das *p-Quantil* den Messwert x_q in der Messreihe, für den mindestens $p \cdot 100\%$ der Messwerte kleiner oder gleich und mindestens $(1-p) \cdot 100\%$ größer oder gleich sind.

Formal: $x_q = \begin{cases} x_{np}, & \textit{falls np ganzzahlig} \\ x_{\lfloor np+1 \rfloor}, & \textit{sonst} \end{cases}$

mit $\lfloor a \rfloor$ = größte ganze Zahl $\leq a$.

Analog wird das *p-Quantil* einer Zufallsvariablen X definiert: Sei F die Verteilungsfunktion von X. Dann heißt die Zahl $x_q = \sup\{x \in \Re : F(x) < p\}$ das *p-Quantil* von X. Falls F streng monoton wachsend und stetig ist, ist x_q eindeutig bestimmt durch $F(x_q) = p$.

Dieses bedeutet für die Messdaten, dass aus ihnen relative Häufigkeiten der Antwortzeiten von Anwendungen ermittelbar sein sollten. Hieraus kann anschließend die zugehörige empirische Verteilungsfunktion ermittelt werden.

Abbildung 03-1 stellt beispielhaft ein Histogramm von Antwortzeiten und die zugehörige empirische Verteilungsfunktion dar. Mit Hilfe der Verteilungsfunktion lässt sich entsprechend des Beispiels die Aussage treffen: „mit 90 %-iger Wahrscheinlichkeit ist die Antwortzeit ≤ 2 Sekunden“. *2* ist also das *0,9-Quantil* der Antwortzeit.

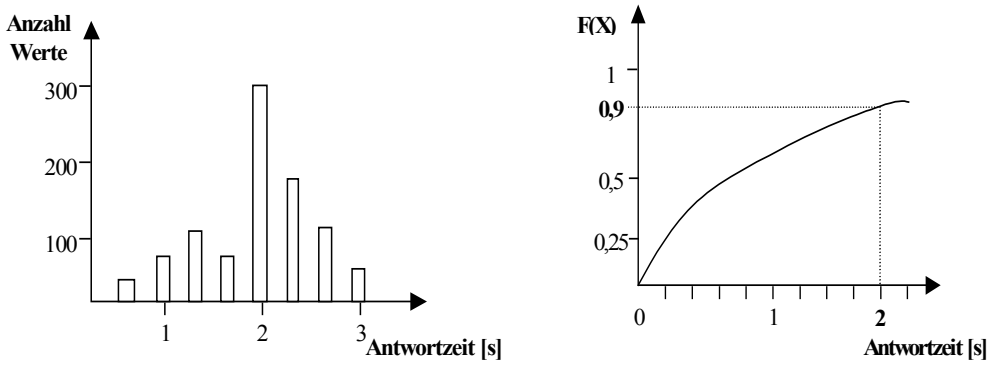


Abbildung 03-1: Beispielhistogramm mit zugehöriger empirischer Verteilungsfunktion

Bei den Messungen muss dafür gesorgt werden, dass die ermittelten Antwortzeiten sich auf einzelne Aufträge beziehen. D.h. es muss sichergestellt sein, dass die gemessenen Start- und Endzeitpunkte einer Anwendungstransaktion tatsächlich zu derselben Transaktion desselben Aufrufs gehören. Mit anderen Worten: es muss eine Zuordnung von Start- und Endzeitpunkten zu einzelnen Transaktionen erfolgen.

(b) Allgemeine Anforderungen an Messungen

Neben den beschriebenen speziellen Anforderungen an die Messdaten, die sich aus der Service-Level-Überwachung ergeben, existieren weitere allgemeine Anforderungen an Messungen, die nachfolgend aufgezeigt werden.

Messung einzelner Aktionen

Wie bereits erwähnt, besteht eine überaus wichtige Anforderung an Messungen in der Möglichkeit der Beobachtung einzelner Benutzeraktionen bzw. Anwendungstransaktionen. Prinzipiell gibt es zwei Möglichkeiten dieses zu erreichen:

@ Messungen einzelner Aktionen im **„leeren" System**: „Leer" bedeutet, dass das System keinerlei Last ausgesetzt wird. In diesem System führt nun ein einzelner Benutzer die Aktionen aus, die vermessen werden sollen. So ist garantiert, dass die Ressourcenanforderungen allein auf die einzeln ausgeführten Aktionen zurückzuführen sind. Ein „leeres" System kann dadurch erzeugt werden, dass im produktiven System der normale Benutzerbetrieb über einen bestimmten Zeitraum hinweg unterbunden wird. Eine weitere Möglichkeit besteht in dem Nachbau des zu untersuchenden Systems in einem Labor.

@ Wenn einzelne Aktionen im **laufenden Betrieb** des Systems vermessen werden sollen, bedeutet dieses, dass eine Art Transaktionverfolgung stattfinden muss. Es muss eine Zuordnung zwischen einer einzelnen Aktion bzw. Transaktion und den Performance-Werten möglich sein. Eine Transaktionsverfolgung erfordert in der Regel die Erweiterung der zu untersuchenden Anwendungen um Identifikationsinformationen für ihre einzelnen Transaktionen.

Beobachtungszeitraum/-intervalle

Der Beobachtungszeitraum, d.h. der Zeitraum oder auch das Zeitfenster der Messung, ist dem Ziel der Systemanalyse anzupassen. Sollen z.B. Aussagen über Wochenprofile von Performance-Werten getroffen werden, so ist mehrmals ein gesamter Arbeitstag zu vermessen.

Weiterhin können aus Gründen der Übersichtlichkeit oder zur Reduktion des Datenvolumens Messdaten innerhalb kleinerer Intervalle aggregiert werden. Die Findung der angemessenen Intervallgröße ist im Laufe der Messungen in der Regel ein iterativer Prozess. Die Kunst besteht darin, die Intervalle so groß zu wählen, dass das Datenvolumen möglichst gering bleibt, sie aber so klein zu wählen, dass ausreichend detaillierte Performance-Aussagen möglich sind.

Synchronizität

Zur Ursachenanalyse bestimmter Performance-Phänomene in dem vermessenen System ist es unerlässlich, dass die Messungen der unterschiedlichen Systembereiche zeitlich synchron ablaufen. Dieses bedeutet nicht nur, dass der Gesamtbeobachtungszeitraum der Messungen übereinstimmen muss, sondern dass darüber hinaus bei einer Unterteilung des Beobachtungszeitraums in einzelne Intervalle, diese ebenfalls übereinstimmen müssen.

(c) Gründe für das Scheitern von Messungen

Grundsätzlich besteht eine Schwierigkeit darin, Messwerkzeuge zu finden, die speziell die gewünschten Daten liefern. Aber auch wenn diese existieren, bestehen weitere Schwierigkeiten in der Erfüllung der allgemeinen Anforderungen:

Ein genereller Grund für das Scheitern von Messungen, die vom Systembetreiber selbst durchgeführt werden, besteht in dem in der Regel stark beschränkten Budget, das ihm zur Performance-Analyse zur Verfügung steht. Derartige Messungen sollen normalerweise „nebenbei“ durchgeführt werden. Sollen Messungen über einen ganzen Tag oder einen längeren Zeitraum hinweg durchgeführt werden, erfordert dieses aber einen erhöhten Aufwand für einen nicht unerheblichen Zeitraum. Weiterhin ist die Bewältigung des großen Datenvolumens ein Problem, wenn die Daten nicht aggregiert werden. Die Entwicklung geeigneter Aggregierungsmethoden erfordert wiederum zusätzlichen Aufwand.

Synchrone Messungen stellen ein weiteres Problem dar, da die meisten Messwerkzeuge synchrone Messungen nicht unterstützen. Die Synchronizität „von Hand“ sicherzustellen, erfordert einen nicht unerheblichen Aufwand.

Die Messung einzelner Transaktionen gestaltet sich schwierig. Ein tatsächlich „leeres“ Produktivsystem steht selten zur Verfügung. Selbst zu Zeiten, in denen die Benutzer nicht aktiv sind, laufen häufig Batch-Programme oder Datensicherungen, die das System belasten. Der Nachbau des Systems in einem Labor ist mit hohem Aufwand verbunden - einmal abgesehen von der Problematik, dass in den wenigsten Fällen ein solches Labor zur Verfügung steht. Weiterhin erfordert der Nachbau detaillierte Systemkenntnisse, die in der Praxis selbst beim Systembetreiber nicht immer anzutreffen sind. Eine (clientseitige) Transaktionsverfolgung kann hier helfen.

03.03 Transaktionsverfolgung mit (Web)Tracy

Die Messung von Einzelaktionen während des laufenden Systembetriebs ist ein lohnender Ansatz, da anders als im „leeren“ System zusätzlich das Performance-Verhalten der Aktionen bei unterschiedlichen Systembelastungen gemessen werden kann. Außerdem verlangt dieser Ansatz keinen speziellen Systemzustand (leeres System). Allerdings erfordert dieses Vorgehen in der Regel eine Modifikation der zu überwachenden Anwendungen, die dann z.B. ihre einzelnen Transaktionen in Form von Traces (Ereignisspuren) protokollieren. Ein solcher Eingriff ist in der Regel nur bei Eigenentwicklungen und nicht bei Standard-Software möglich.

Eine Besonderheit stellt die Transaktionsverfolgung von Client-Server-Anwendungen dar. Jede Systemkomponente, die von der Anwendung benutzt wird, müsste die Verweildauer und die Ressourcen-Verbräuche einer Transaktion protokollieren. Die Schwierigkeit besteht hierbei in der eindeutigen Identifikation der Transaktionen, da sie oft in weitere Teiltransaktionen zerfallen. So ist z.B. eine Zuordnung von einzelnen Datenbankanfragen zu der sie erzeugenden Transaktion kaum möglich.

Es ist daher oft ratsam und ausreichend, sich auf eine clientseitige Transaktionsverfolgung zu beschränken, bei der die Aktionen, die auf dem Client-Rechner ausgeführt werden, protokolliert werden, während das umgebende System als Black-Box betrachtet wird, vgl. Abbildung 03-2.

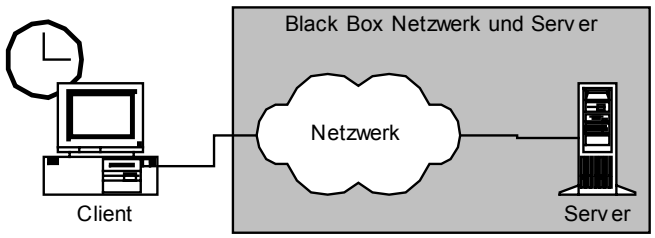


Abbildung 03-2: Prinzip der clientseitigen Tranksaktionsverfolgung

Die Performance-Daten werden clientseitig gemessen und analysiert. Bei der Zusammensetzung der Werte kann nach Client- und „Rest"-Performance (Netz und Server) unterschieden werden.

(a) Tracy / WebTracy

Tracy ist ein an der Universität Essen entwickelter Offline-Analysator, d.h. ein Programm zur Offline-Analyse der Performance von Client/Server-Anwendungen. Es ist nicht auf einen bestimmten Anwendungstyp beschränkt. Die Voraussetzung ist eine **instrumentierte** Anwendung, die Ereignis-Traces in einem für Tracy lesbaren Format erzeugt. Die Instrumentierung der Anwendung erfordert einen relativ geringen Eingriff in den Source-Code, da sie keinerlei Berechnungen von Performance-Werten beinhaltet, sondern lediglich die Protokollierung einzelner Ereignisse mit dazugehörigen Antwortzeiten/Zeitstempeln und ggf. zusätzlichen Informationen wie z.B. transferiertes Datenvolumen gewährleisten soll, vgl. Abbildung 03-3.

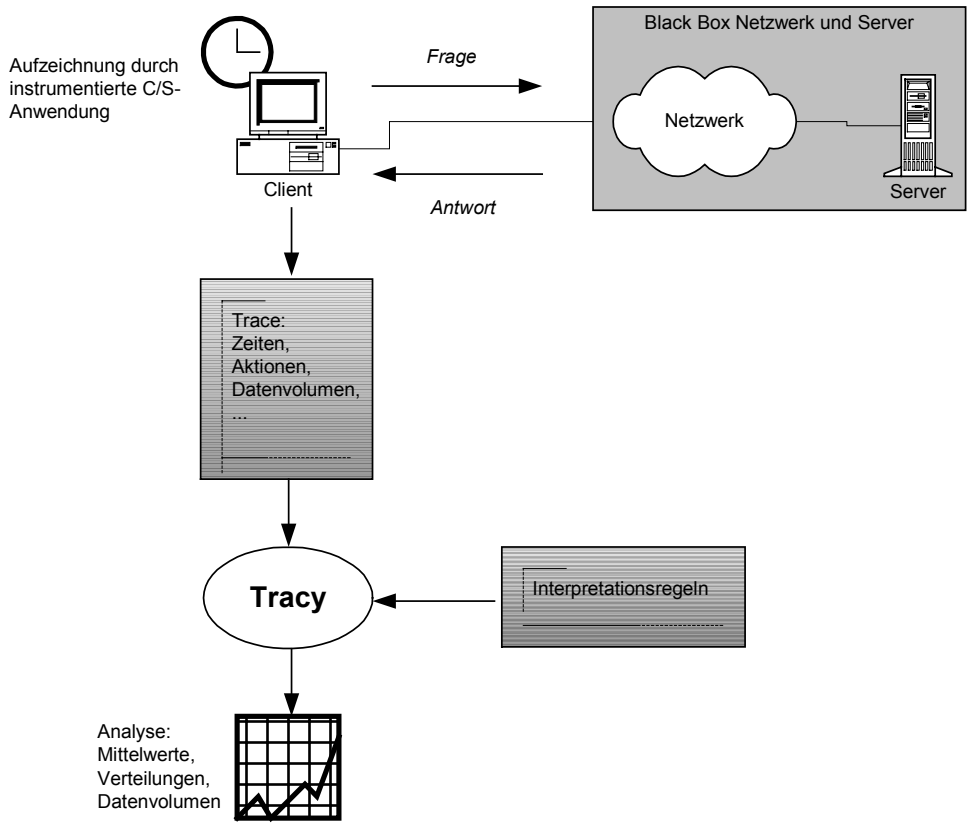


Abbildung 03-3: Tracy-Konzept

Für weitere Informationen zu Tracy siehe [2] und [3].

WebTracy ist eine Variante von Tracy, die für die Analyse von Webseiten optimiert ist. WebTracy kann Traces analysieren, die von dem Web-Browser *WebChecker* (vgl. [6]) erzeugt wurden. WebChecker ist ein instrumentierter, textbasierter Web-Browser für LINUX-Systeme, der an der Universität Essen entwickelt wurde. Er misst die Ladezeiten/Antwortzeiten und das Datenvolumen von Webseiten und deren Bestandteile (Bilder, Applets etc.), wobei Client-Verarbeitungszeiten nicht berücksichtigt werden.

WebTracy erzeugt aus den WebChecker-Traces folgende Analyseergebnisse:

@ Eine Übersicht über die Objekte, aus der sich die Web-Seite zusammensetzt, ihre Vorkommenshäufigkeit und ihr Datenvolumen in Byte,

- ein Histogramm aus den Objektladezeiten (relative Häufigkeiten von Antwortzeiten, kumulierte Häufigkeiten von Antwortzeiten → Antwortzeitquantile),
- Mittelwert, Standardabweichung und min./max.-Werte der Antwortzeiten,
- absolute Antwortzeiten.

Es können einzelne Objekte der Seite oder die gesamte Seite analysiert werden. WebTracy besitzt eine graphische Benutzeroberfläche, von der aus auch WebChecker bedienbar ist (vgl. Abbildung 03-4).

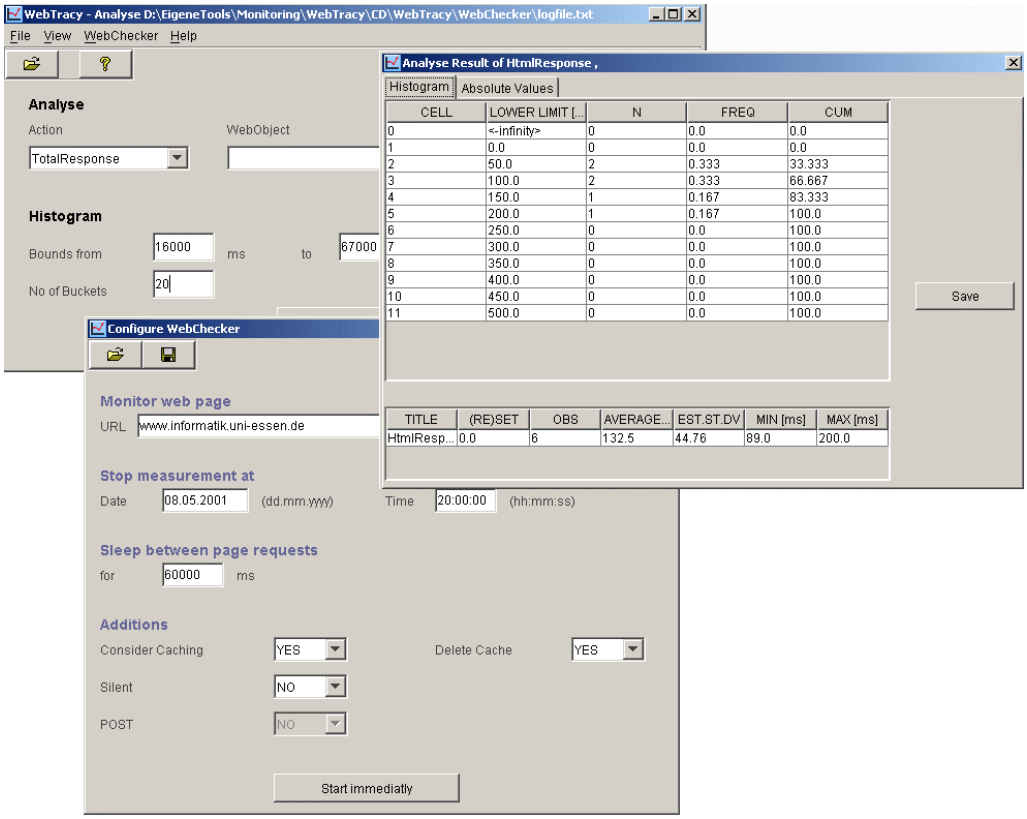

WebTracy - Analyse D:\EigeneTools\Monitoring\WebTracy\CD\WebTracy\WebChecker\logfile.txt
File View WebChecker Help
Analyse
Action
WebObject
TotalResponse
Histogram
Bounds from 16000 ms to 67000
No of Buckets 20
Analyse Result of HtmlResponse ,
Histogram Absolute Values
CELL | LOWER LIMIT [... | N | FREQ | CUM
0 | <-infinity> | 0 | 0.0 | 0.0
1 | 0.0 | 0 | 0.0 | 0.0
2 | 50.0 | 2 | 0.333 | 33.333
3 | 100.0 | 2 | 0.333 | 66.667
4 | 150.0 | 1 | 0.167 | 83.333
5 | 200.0 | 1 | 0.167 | 100.0
6 | 250.0 | 0 | 0.0 | 100.0
7 | 300.0 | 0 | 0.0 | 100.0
8 | 350.0 | 0 | 0.0 | 100.0
9 | 400.0 | 0 | 0.0 | 100.0
10 | 450.0 | 0 | 0.0 | 100.0
11 | 500.0 | 0 | 0.0 | 100.0
Save
TITLE | (RE)SET | OBS | AVERAGE... | EST.ST.DV | MIN [ms] | MAX [ms]
HtmlResp... | 0.0 | 6 | 132.5 | 44.76 | 89.0 | 200.0
Configure WebChecker
Monitor web page
URL www.informatik.uni-essen.de
Stop measurement at
Date 08.05.2001 (dd.mm.yyyy)
Time 20:00:00 (hh:mm:ss)
Sleep between page requests
for 60000 ms
Additions
Consider Caching YES
Delete Cache YES
Silent NO
POST NO
Start immediatly

Abbildung 03-4: WebTracy-Oberfläche

(b) Abgrenzung Tracy - ARM

Das Konzept von Tracy weist Ähnlichkeiten zu dem des *Application Response Monitoring* (ARM) auf. Es bestehen jedoch folgende Unterschiede:

@ ARM führt eine Online-Analyse durch, d.h. die Instrumentierung („ARMierung“) der Anwendung beinhaltet bereits umfangreiche Berechnungsalgorithmen zur Performance-Analyse. Die Instrumentierung ist also komplexer als bei Tracy, wodurch die Beeinträchtigung der Anwendungs-Performance entsprechend größer ist. Andererseits sind die Ergebnisdaten bereits durch die Analyse verdichtet, während das Datenaufkommen bei Tracy größer ist, da die instrumentierte Anwendung Einzelereignisse protokolliert.

@ Bei ARM gibt es keine Trennung zwischen Aufzeichnung und Analyse.

@ ARM ist auf eine Transaktionsverfolgung im eigentlichen Sinne ausgerichtet, d.h. jede Transaktion wird mit einer Kennung versehen und auf ihrem gesamten Weg durch das System verfolgt. Tracy ist für den Einsatz des clientseitigen Monitorings konzipiert. Theoretisch könnten zwar Ereignis-Traces von allen beteiligten Systemkomponenten erzeugt werden, in die Anwendungsinstrumentierung müssten dann allerdings u.a. Mechanismen zur Zeitsynchronisation der Komponenten integriert werden, was einen erheblichen Aufwand darstellt.

03.04 Fallstudie

In der Fallstudie wird ein Online-Banking-Service einer Bank überwacht. Stichproben haben gezeigt, dass die Antwortzeiten z.T. über dem noch tolerierbaren Schwellenwert von 8 Sekunden liegen. Gezielte Messungen sollen die Einhaltung des Service-Levels von 8 Sekunden bei 90 % der gemessenen Antwortzeiten überprüfen.

(a) Messkonzept

WebChecker greift von einem Linux-PC der Universität Essen aus auf die Einstiegswebseite für das Online-Banking zu. Der PC ist über das Wissenschaftsnetz mit dem Internet verbunden, es ist kein zusätzlicher Internet-Provider beteiligt. Abbildung 03-5 stellt die Messumgebung graphisch dar.

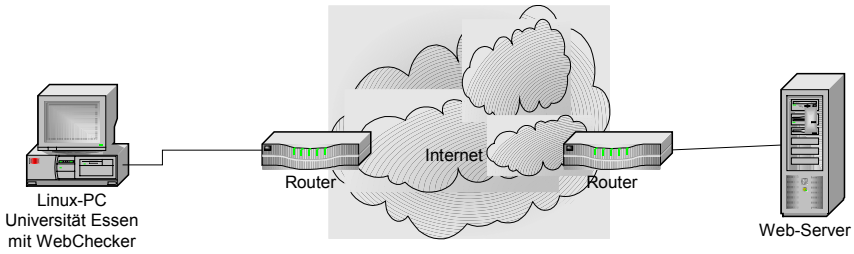


Abbildung 03-5: Messumgebung

Messung einzelner Aktionen

Da die Performance-Werte der Web-Seite bei unterschiedlichen Systembelastungen gemessen werden sollen, werden Messungen im laufenden Betrieb vorgenommen. Da kein Zugriff auf das interne IT-System der Bank besteht, wird eine clientseitige Transaktionsverfolgung mit WebChecker und WebTracy durchgeführt. Das Caching von WebChecker wird deaktiviert, d.h. ein Request hat stets das erneute Laden der Seite vom Web-Server über das Internet zur Folge. WebChecker protokolliert in einer Trace-Datei die Zeit, die im Web-Server und im Internet dabei verbraucht wird. Verarbeitungszeiten auf dem Client-PC werden nicht berücksichtigt.

WebTracy liefert als Analyseergebnis Mittelwerte wie auch Antwortzeitverteilungen, d.h. Antwortzeitquantile. Somit lässt sich die Service-Level-Einhaltung überprüfen.

Beobachtungszeitraum/-intervalle

Bankkunden greifen üblicherweise vermehrt während ihrer Freizeit auf Online-Banking-Seiten zu. Spezielle Online-Tarife, die ein kostenloses Surfen an Sonn- und Feiertagen bieten, erhöhen die Zugriffszahlen auf Web-Server am Wochenende. Daher wird die Einstiegsseite für das Online-Banking während eines Wochenendes mit WebChecker beobachtet, d.h. von Freitag Abend bis Montag Vormittag. Während dieses Zeitraums wird die Web-Seite alle 10 Minuten von WebChecker geladen.

(b) Überwachung der Service Level Agreements

Insbesondere bei kommerziellen Web-Seiten sollte die „8-Sekunden-Regel“ als Service Level Agreement gelten: die Antwortzeit der Web-Seite sollte in 90 % der Fälle 8 Sekunden nicht übersteigen, d.h. das 90 % -Quantil sollte bei 8 Sekunden liegen.

Die WebChecker-Trace-Dateien werden mit WebTracy ausgewertet. Abbildung 03-6 zeigt die resultierenden Antwortzeitquantile. 90 % der gemessenen Antwortzeiten liegen unterhalb von 18,6 Sekunden. Der Service Level wird um den Faktor 2,3 überschritten.

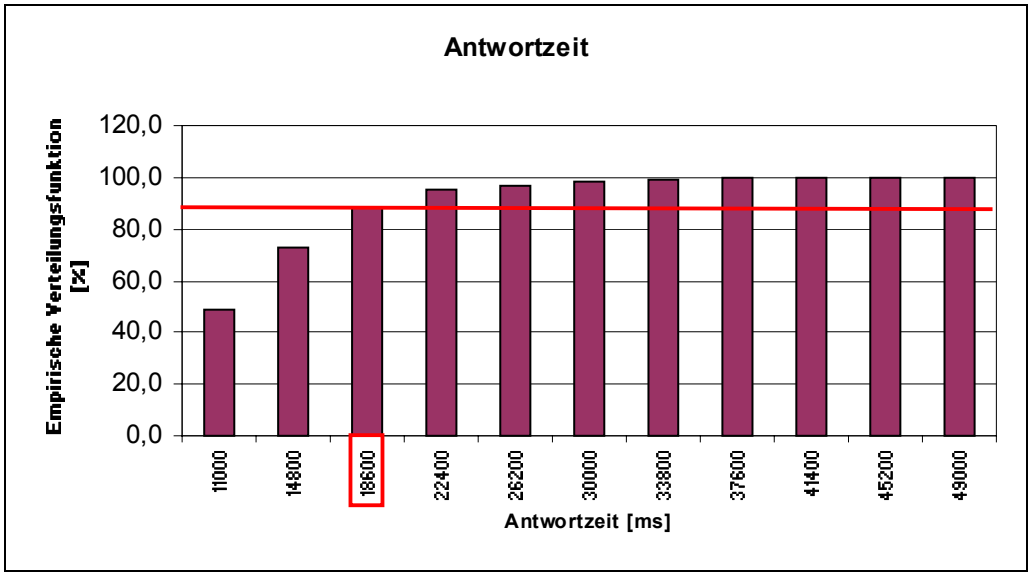


Abbildung 03-6: Antwortzeitquantile

(c) Analyse

Abbildung 03-7 zeigt die absoluten Antwortzeiten für die gesamte Web-Seite entlang der Zeitachse. Die mittlere Antwortzeit von 16,4 Sekunden (vgl. auch Tablelle 03-1) ist zu hoch. Die höchsten Antwortzeiten werden am beobachteten Samstag zwischen 11 Uhr und 19.40 Uhr erreicht. Noch höhere Werte sind am Sonntag zwischen 12.30 Uhr und 20.40 Uhr zu beobachten. In diesem Zeitintervall wird der Maximalwert von 49 Sekunden erreicht.

Insgesamt ist festzustellen, dass zu keinem Zeitpunkt innerhalb des beobachteten Zeitraums die geforderte Antwortzeit von 8 Sekunden erreicht wird. Es wird ein Minimalwert von 11

Sekunden erreicht. Darüber hinaus streuen die gemessenen Antwortzeiten recht stark (vgl. Standardabweichung – EST.ST.DV. in Tabelle 03-1).

AVERAGE [ms]	EST.ST.DV	MIN [ms]	MAX [ms]
16438	5399	11000	49000

Tabelle 03–1: Erstes und zweites Moment, Minimum und Maximum der Antwortzeit in Millisekunden

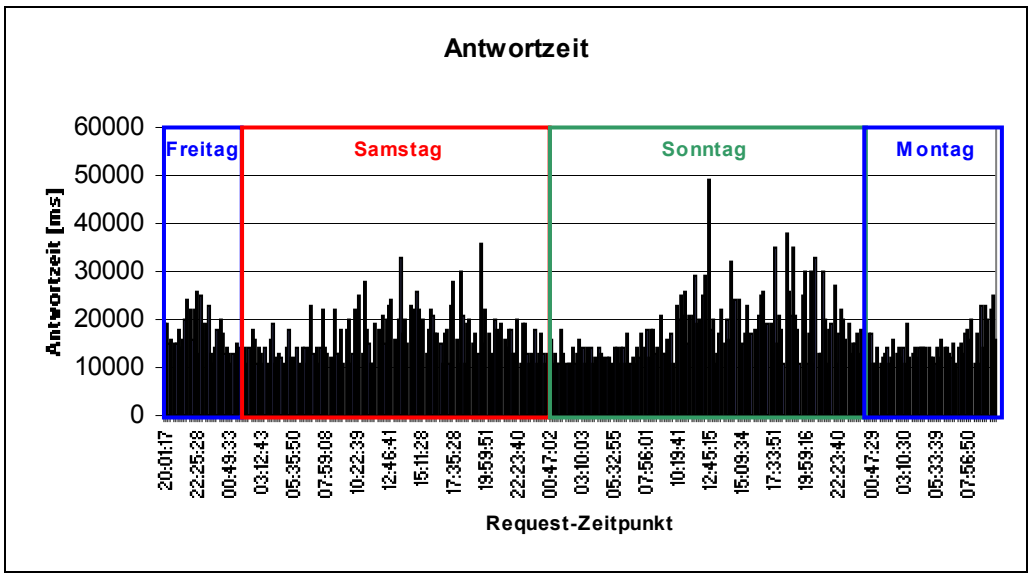


Abbildung 03-7: Absolute Antwortzeiten (Gesamt-Frame)

Für die Ursachenanalyse empfiehlt es sich, zunächst die Struktur der Web-Seite zu betrachten. Abbildung 03-8 zeigt ihre Zusammensetzung. Die Seite setzt sich aus 28 Web-Objekten zusammen und besitzt ein Datenvolumen von insgesamt 25,502 Kilobytes.

Abbildung 03-9 stellt den sich daraus ergebenden Durchsatz beim Laden der Web-Seite entlang der Zeitachse dar. Es werden Höchstwerte von 2,3 Kbyte/s erreicht.

Das Bild „l_banner486x60.gif“ umfasst mit 9,289 Kilobyte das größte Datenvolumen innerhalb der Web-Seite. Abbildung 03-10 zeigt die relativen Häufigkeiten der Ladezeit für dieses Bild. Sie liegen zu etwa 50 % zwischen 9,6 und 13 Sekunden. Der Maximalwert liegt bei 43,4 Sekunden. D.h. bereits die Ladezeiten dieses einzelnen Objekts sind recht hoch.

Die Ladezeiten für den reinen HTML-Code, der 8,358 Kilobytes umfasst, sind in Abbildung 03-11 dargestellt. Etwa 50 % der Ladezeiten liegen zwischen 10,5 und 14,3 Sekunden, was ebenfalls recht hoch ist.

(d) Fazit

Zusammenfassend ist festzustellen, dass die Ursachen für die Performance-Probleme offensichtlich nicht in der Web-Seite an sich begründet sind. Die Anzahl von 28 Objekten ist für eine Web-Seite zwar recht hoch, das Datenvolumen liegt aber im normalen Bereich. Vielmehr sind die zu beobachtenden Ladezeiten und der Durchsatz selbst für einzelne Objekte zu schlecht. Aufgrund des längeren Beobachtungszeitraums und aufgrund der Tatsache, dass Datenpakete in der Regel auf unterschiedlichen Wegen durch das Internet geroutet werden, liegt der Schluss nahe, dass die Ursache für das Performance-Problem nicht im Internet selbst, sondern „hinter“ dem Internetzugangspunkt der Bank liegt. Möglicherweise handelt es sich um ein Kapazitätsproblem des Web-Servers. Um derartige Aussagen verifizieren zu können, müssten synchron zu den durchgeführten clientseitigen Messungen Messungen im System der Bank vorgenommen werden. Dieses ist innerhalb der Fallstudie nicht möglich, da kein Zugriff auf das interne System der Bank besteht.

	Number	Size [Byte]
testkonto.gif	1	486
sat-preview.gif	1	1069
spacer.gif	9	43
foot_right.gif	1	56
ecoupon-ok.gif	1	201
ecoupon-abbrechen.gif	1	327
foot_left.gif	1	56
LogIn.gif	1	328
pixel_weiss.gif	3	43
l_deutschepost.gif	1	1433
l_banner468x60.gif	1	9289
broking.gif	1	344
home.gif	1	199
pblogo.gif	1	1139
bg.gif	1	1701
anfang.jsp	1	7669
empty.html	1	152
https://www.testbank.de	1	537

Abbildung 03-8: Zusammensetzung der Web-Seite
(WebTracy-Menü „Objects in web page“)

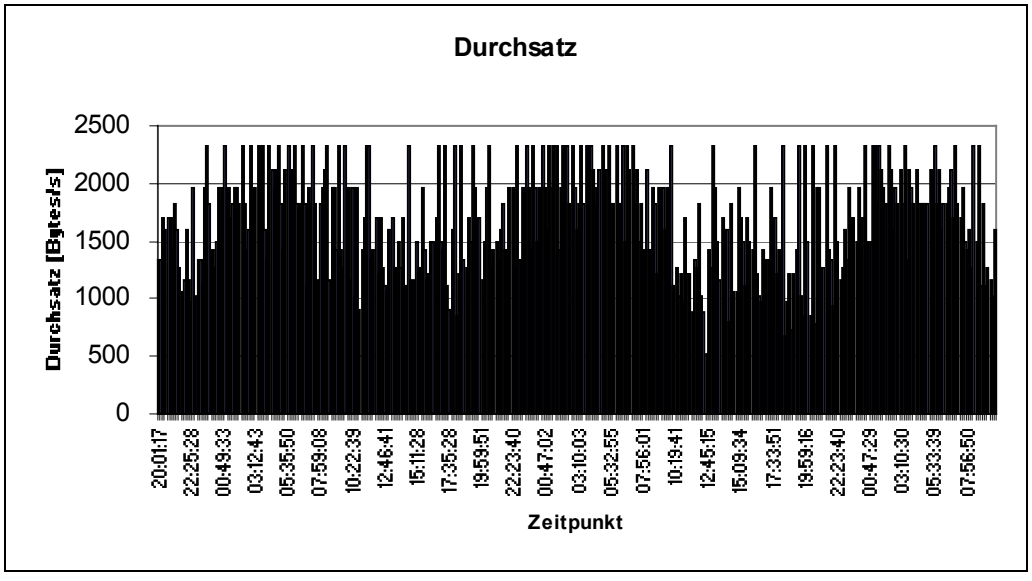


Abbildung 03-9: Durchsatz beim Laden der Web-Seite

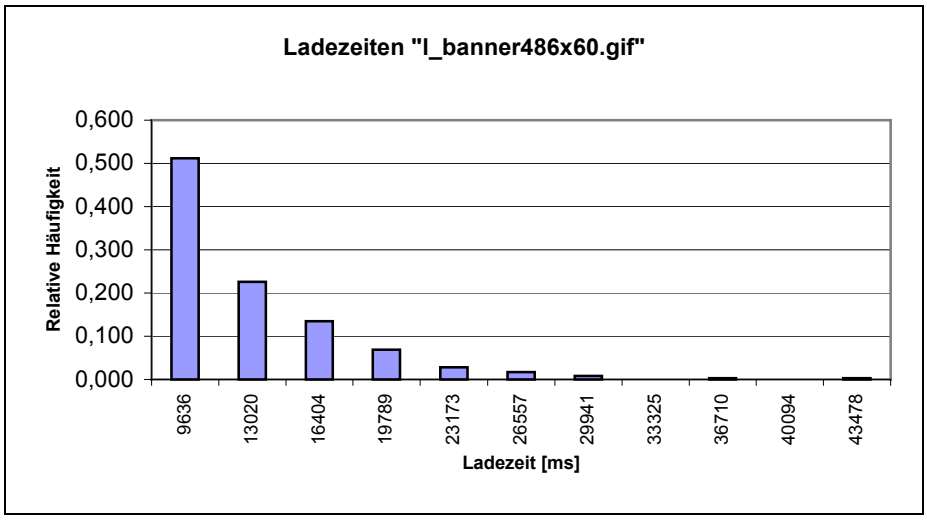


Abbildung 03-10: Ladezeit des größten Bildes der Web-Seite (Histogramm)

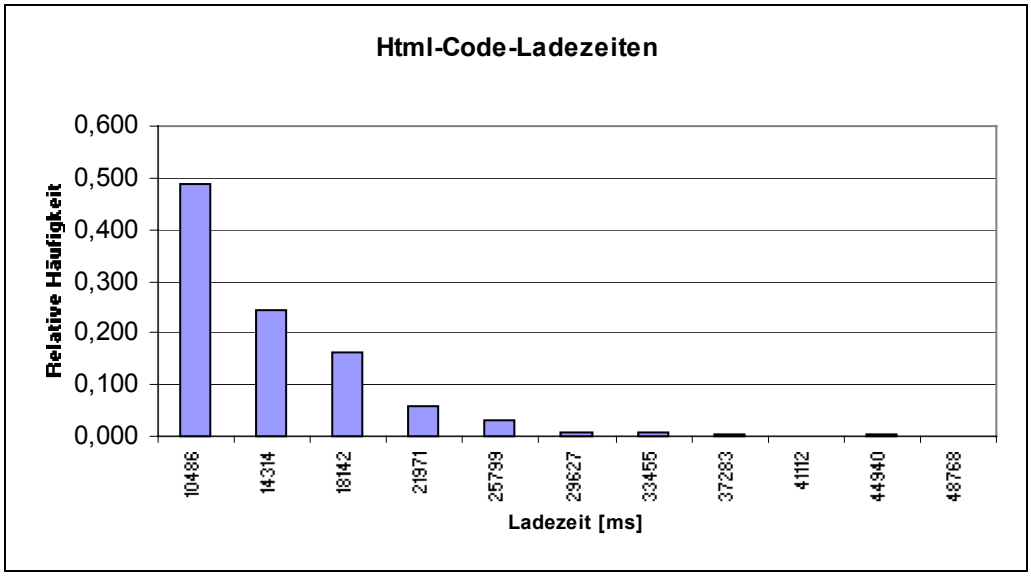


Abbildung 03-11: Ladezeiten für den HTML-Code (Histogramm)

03.05 Ergänzende Literatur

[1] J. Lehn, H. Wegmann: *Einführung in die Statistik*. Teubner Studienbücher Mathematik. B. G. Teubner, Stuttgart, 2. Edition, 1992.

[2] C. Flüs*: WebTracy User Guide*. Universität Essen, Mai 2001.

[3] M. Vilents, C. Flüs, B. Müller-Clostermann: *VITO: A Tool for Capacity Planning of Computer Systems*. MMB'99, 10. GI/ITG-Fachtagung 22.-24.09.1999 in Trier. Forschungsbericht Nr. 99-17, Kurzbeiträge und Toolbeschreibungen.

[4] VITO-Homepage: http://www.cs.uni-essen.de/VITO.

[5] C. Flüs: *Trace-basierte Performance-Analyse einer Client/Server-Anwendung*. Technischer Bericht Nr. 3, Universität Essen, Mai 2000.

[6] S. Laaks: *WebChecker User Guide*. Universität Essen, Mai 2001.

Teil VI: SAP R/3

Inhalt von Teil VI

KAPITEL 01 SAP R/3 KAPAZITÄTSMANAGEMENT – GRUNDLAGEN UND WERKZEUGE IM ÜBERBLICK 3

01.01 EINFÜHRUNG UND AUFGABENSTELLUNG 3
01.02 LÖSUNGSKONZEPT 3
01.03 EIN TYPISCHES R/3 PROJEKTSZENARIO 5
01.04 DIE WERKZEUGE UND VERFAHREN 6
01.05 ZUSAMMENFASSUNG 15
01.06 LITERATUR UND REFERENZEN 15

KAPITEL 02 MONITORING UND BENCHMARKING FÜR DAS R/3-PERFORMANCE-ENGINEERING 16

02.01 EINLEITUNG 16
02.02 SAP R/3 – EIN STANDARD-SOFTWARE-PAKET 17
02.03 MESSKONZEPT FÜR R/3-LAST 19
02.04 MONITORING-WERKZEUGE 22
02.05 SAP STANDARD APPLIKATIONS BENCHMARKS 33
02.06 BEISPIELE FÜR IST-ANALYSEN 43
02.07 AUSBLICK 46
02.08 LITERATUR 46

KAPITEL 03 MODELLIERUNG UND PROGNOSE FÜR SAP R/3-SYSTEME MIT DEM WLPSIZER 48

03.01 EINLEITUNG 48
03.02 MODELLIERUNGSWERKZEUG WLPSIZER - ARCHITEKTUR 48
03.03 WLPSIZER IM ÜBERBLICK 61
03.04 MODELL-EVALUATION 74
03.05 ERGEBNISSE 74
03.06 ERWEITERTE FUNKTIONEN UND VORGEHENSWEISEN ZUR MODELLIERUNG 82
03.07 BEISPIELE 94
03.08 LITERATUR 101

KAPITEL 01 SAP R/3 KAPAZITÄTSMANAGEMENT – GRUNDLAGEN UND WERKZEUGE IM ÜBERBLICK

MICHAEL PAUL, KAY WILHELM

01.01 Einführung und Aufgabenstellung

Als Lieferant von Systemplattformen für SAP R/3 Anwendungen stehen wir[1] regelmäßig vor der Aufgabe, das Leistungsverhalten des Systems unter der geplanten Last des Kunden zu beschreiben und zu garantieren. Dieses „Sizing" wird von allen Beteiligten als „riskant" beschrieben, weil in aller Regel zum Zeitpunkt des Sizings weder die konkrete Ausprägung der Last durch Customizing und Nutzungsgewohnheiten festliegt, noch Mengengerüst und Batch-Produktionsplan die notwendige Reife haben.

Das Problem besteht darin, für ein bewegliches Ziel Performanceaussagen zu machen, die nicht „leer" sind. Oft findet man Klauseln wie: „*Die Angaben des Sizing Tools bestimmen die HW-Konfiguration genau, wenn das Lastprofil des "Realen Users" dem des "Benchmark Users" ähnlich ist. Abweichungen der R/3 Lasten sind durch das R/3 Customizing und andere Einflussfaktoren möglich*." Bei nicht erfüllten Leistungserwartungen des Kunden ist der Ärger vorprogrammiert und das Performanceproblem erscheint in der Regel als Mangel der Konfiguration und nicht des Projektprozesses.

01.02 Lösungskonzept

Mit unserem Ansatz zum SAP R/3 Kapazitätsmanagement lösen wir die Aufgabe, eine Last in allen Projektphasen von der Planung bis zum Produktionseinsatz überprüfbar zu beschreiben. Als gemeinsame Darstellungsform für die Messung und Modellierung verwenden wir auf DB-Service basierte Komplexitätsklassen. In dem Beitrag werden die methodischen und technischen Grundlagen, ein konkretes Projekt sowie die eingesetzten Werkzeuge beschrieben.

Für die Absicherung des Projekterfolges des Kunden bei Performance-kritischen Anwendungen und zur Versachlichung der Diskussion zwischen den „Parteien" Kunde, Realisierungs-Team und Plattformlieferant haben wir unter der Bezeichnung R/3 Ka-

[1] Fujitsu Siemens Computers GmbH

pazitätsmanagement ein Verfahren entwickelt und eingeführt, das es erlaubt, die geplante Last, die als Prototyp im Rahmen von Integrationstests messbare Last und die produktive Last aufeinander zu beziehen und zu vergleichen. Der besondere Charme des Vorgehens besteht darin, dass es Branchen-unabhängig sowohl für Standard-Dialogschritte als auch für Eigenentwicklungen anwendbar ist.

Die Lösung benutzt die Methoden zur Beschreibung und Performanceanalyse von Client-Server-Systemen mit effizient behandelbaren mathematisch-analytischen Modellen. Für die Modellbildung und den Vergleich des Modells mit der Wirklichkeit werden Lastprofile toolbasiert erhoben. Bei der Profilbildung werden als elementare Konzepte *Klassenbildung nach Komplexitätsklassen* und *komplexitätsabhängige Servicegüte-Bewertung* verwendet.

(a) Komplexitätsklassen für R/3-Dialogschritte

Es wird ein skalares Komplexitätsmaß für Dialogschritte eingeführt, das von der verwendeten Hardware-Plattform, dem R/3-Release, der verwendeten Systemsoftware unabhängig ist. Als Maß für die Komplexität eines Dialogschrittes werden *DB-Service-Units* verwendet. Sie sind realisiert als gewichtete Summe über die Datenbankaktivitäten eines Dialogschrittes. Die Rohdaten hierfür befinden sich in den Statistik-(Account)Sätzen, die zum R/3 Basisfunktionsumfang gehören. Die DB-Service-Units als Maß für die Komplexität eines Dialogschritts eignen sich gut zur Verständigung zwischen Anwendern, Entwicklern und Systemleuten, weil sie intuitiv verständlich sind. „Entartungen“ von Dialogschritten (z.B. viel CPU-Verbrauch bei geringer DB-Aktivität) können leicht erkannt und auf Plausibilität überprüft werden.

Die Komplexitätsklassen beruhen auf DB-Service, sind unabhängig von der Applikation, der Hardware-Plattform und auch unabhängig vom Betriebssystem und der Datenbank-Software. Sie repräsentieren die Komplexität des Dialogschritts in 5 Klassen von „sehr einfach“ (100 DB-Service-Units) bis „sehr komplex“ (über 100.000 DB-Service-Units). Die Komplexitätsklassen werden berechnet für die Tasktypen Dialog, Update, Batch und Other.

(b) Servicegüteklassen für die Antwortzeit

Eine weitere Säule des Vorgehens ist die Definition und Messmethode von Servicegüte. Schon lange hat sich herumgesprochen, dass die mittlere Antwortzeit des Dialogschritts kein aussagefähiges Maß für die Güte des Zeitverhaltens ist. Gerade in R/3 kann man mit wenigen Handgriffen eine hohe Anzahl von Dialogschritten mit sehr gutem Zeitverhalten erzeugen, die den Mittelwert als Bewertungskriterium unbrauchbar machen. Unser Ansatz bewertet die Antwortzeit im Verhältnis zur Komplexität des Dialogschritts. Mit einer einfachen Formel wird eine ”gleitende Antwortzeitforderung” definiert. So kann mit den Güte-

attributen ”gut”, ”mäßig” und ”schlecht” ein zutreffendes Bild der Servicegüte erzielt werden.

Servicegüteklassen sind realisiert als Schwellwerte pro Komplexitäts-Klasse, sie reichen von sehr einfach (250 msec) , über einfach (1000 msec) etc. bis zu sehr komplex (1 msec pro DB-Service-Unit). Daraus resultiert eine komplexitätsabhängige Antwortzeit-Bewertung in den drei Stufen „gut“ (Messwert <= Schwellwert), „mäßig“ (Messwert <= 4*Schwellwert) und „schlecht“ (Messwert > 4*Schwellwert).

01.03 Ein typisches R/3 Projektszenario

Bei einem unserer großen Kunden soll das zentrale Mainframe-basierte System durch die SAP R/3 Branchenlösung abgelöst werden. Die Planungen und Architekturüberlegungen werden intensiviert seit 1998 bearbeitet. Das Projekt teilt sich in zwei Phasen der Realisierung, von der für die erste Phase 4 Jahre geplant sind. Auf Grund der Heterogenität der einzelnen Anforderungen ist es geplant, ein verteiltes R/3-Szenario aufzubauen. Die Abstimmung der Aufteilung erfolgt in engster Zusammenarbeit mit der Branchen-Architekturgruppe der SAP in Walldorf. Da eine solche Konfiguration bisher weder durch die SAP noch durch andere realisiert wurde, sind sowohl hinsichtlich der Prozessgestaltung als auch der Lastverteilung viele Entscheidungen noch nicht getroffen und manche Lösung ist noch nicht implementiert. In dem Marktsegment sind sehr hohe Verfügbarkeitsforderungen an die Anwendungen zu stellen. Eine „Einschwingphase“ für die Anwendung unter Produktionsbedingungen ist nicht durchzuhalten, sondern die Anwendung muss „vom Start weg“ die Durchsatz-, Antwortzeit- und Verfügbarkeitsforderungen erfüllen. Das Teilprojekt Kapazitätsmanagement ist eng verknüpft mit dem übrigen Projektgeschehen: Architektur, Hardwarebereitstellung, Systemeinrichtung und -Tuning. Es wurde ein schrittweises Vorgehen vereinbart:

- Abbildung der im Sizing bewerteten Last in ein Lastprofil im vorgestellten Sinne und daraus Erstellen eines Referenzmodells
- Ermitteln des „erwarteten Dialoglastprofils“ aus dem Integrationstest. Hier hatten die künftigen Anwender die Aufgabe, „ein Zehntel des typischen Tagewerks“ in einer definierten Zeitspanne (ein Nachmittag) durchzuführen
- Ermitteln der Ressourcenbedarfe und Laufzeiten für Schnittstellenprogramme zum Datenaustausch mit Partnersystemen (R/3 und Non-R/3)
- Ermitteln der Ressourcenbedarfe und Laufzeiten für Batchläufe

Aus den Messungen wurden nicht nur Eingaben für die Modellierung abgeleitet, sondern vor allem auch proaktiv Analysen und Betrachtungen über die „Berechtigung“ des Ressourcenverbrauchs eingeleitet.

01.04 Die Werkzeuge und Verfahren

Von wesentlicher Bedeutung sind die Werkzeuge R/3 Live Monitor (= myAMC.LNI) und WLPSizer. Der R/3 Live Monitor wird zur Messung von R/3-Lasten im Produktivbetrieb und der WLPSizer für Modellierung und Prognose von R/3-Systemen basierend auf den gemessenen Daten verwendet.

(a) Der R/3 Live Monitor (myAMC.LNI[2]) im Überblick

Das von Siemens entwickelte Werkzeug „R/3 Live Monitor Expert-Management“ [1] ist ein Enterprise IT Management Tool für SAP R/3-Systeme. Es können mit diesem Werkzeug verschiedene SAP R/3-Systeme von einer zentralen Stelle aus überwacht werden (Single Point of R/3 Control). Folgende Eigenschaften sind wesentliche Unterscheidungsmerkmale:

- Übersichtliche Darstellung der Beziehungen, Verteilung und Aktivitäten von R/3-Datenbanken und ihrer zugehörigen R/3-Applikations-Instanzen
- Komplexitäts- und dienstgütebezogene Erfassung von R/3-Last basierend auf Dialogschritten
- Erkennung und Zuordnung von Performanceproblemen in einem R/3-System
- Anzeige von SAP R/3-Alerts (Alarmautomatik)

Die bislang beschriebenen Eigenschaften des R/3 Live Monitor bilden das „R/3 Live Monitor Basis-Management“. Eine Erweiterung stellt das „R/3 Live Monitor Expert-Management“ dar, das zusätzlich zu den bisherigen Basis-Management-Eigenschaften die Komponente „LNISCA[3]“ beinhaltet. Die Erweiterung des R/3 Live Monitor liegt in der Analyse der in einem SAP R/3-System anfallenden Messdaten, die das System standardmäßig für jede Transaktion in Form von Dialogschritten protokolliert.

LNISCA bietet neue Funktionalitäten für SAP R/3 Service Quality, Capacity und Accounting Management und besitzt die folgenden Hauptaufgaben:

---

[2] myAMC.LNI ist die neue Bezeichnung für den R/3 Live Monitor (seit Anfang 2001).
[3] SCA = Service Quality, Capacity and Accounting

- R/3 Workload Management
- R/3 Database Service
- R/3 Service Quality
- User Activity Measurement
- Tracing Significant Transactions
- Accounting Management
- Exception Logging

R/3 Live Monitor Service Quality, Capacity und Accounting (myAMC.LNISCA)

SAP R/3 ist eine transaktionsorientierte Anwendung. Alle durchgeführten Transaktionen werden vom R/3-System protokolliert. Die Transaktionen werden mit Hilfe der vom Dispatcher verwalteten Workprozesse abgearbeitet, die in vier Tasktypen eingeteilt werden:

- Dialog (Dialogverarbeitung)
- Update (Verbuchungen)
- Batch (Hintergrundverarbeitung)
- Others (Spool, Enqeue etc.)

Der Dialog-Anteil einer Transaktion wird in Dialogschritte[4] unterteilt. Jeder Dialogschritt wird durch einen Dialog-Workprozess bearbeitet. Die Verbuchungen einer Transaktion werden in Verbuchungskomponenten aufgeteilt, die den Verbuchungs-Workprozessen zugewiesen werden. Batch-Jobs werden den Background-Workprozessen zur Verarbeitung übergeben. Im Folgenden wird jegliche Durchführung eines Workprozesses als Dialogschritt[5] bezeichnet.

Die LNISCA-Komponente verwaltet eine Liste aller in einem Messzeitraum protokollier_ten Dialogschritte, die im R/3-System ausgeführt wurden. Jedem Dialogschritt werden 88 Attribute zugeordnet, wie zum Beispiel System, Instance, Responsetime, CPU-Time und Database Activity.

---

[4] Ein Dialogschritt entspricht der Verarbeitung eines Bildschirms bzw. einer Eingabemaske.

[5] Diese Sprachregelung wurde von der SAP geprägt. Es besteht somit eine atomare Einheit über alle Tasktypen, jedoch kann es hierdurch zu Verwirrungen führen, da Update, Batch und Others keine Bildschirmwechsel besitzen, wie die Bezeichnung vielleicht vermuten lässt.

Zur Darstellung der gemessenen Dialogschritte bietet der R/3 Live Monitor verschiedene Sichten auf die Daten. Es werden im Folgenden die TCQ-Profile, User-Activity-Profile und Significant Dialogsteps als die drei wichtigsten Darstellungsformen beschrieben.

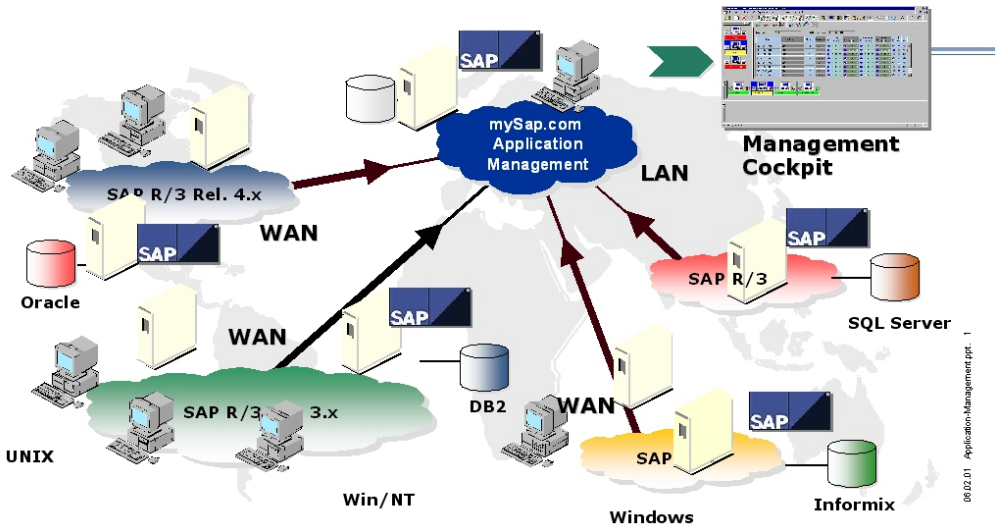


Abbildung 01-1: Management Cockpit des Application Management Centers myAMC.LNI[6]

TCQ-Profile (T60-Tabelle)

Zur Reduktion der Datenmenge werden die gemessenen Dialogschritte aggregiert. Als Ergebnis erhält man die „T60-Tabellen“, deren Zeilen keine einzelnen Dialogschritte, sondern eine Gruppe der in einem bestimmten Zeitzyklus angefallenen Dialogschritte repräsentieren. Die Attribute der T60-Tabelle entsprechen den aufsummierten Verbräuchen aller Dialogschritte einer Gruppe.

In den T60-Tabellen werden die Dialogschritte ergänzend zu den vier Tasktypen in fünf Komplexitätsklassen und drei Service-Güteklassen unterteilt, wodurch auch die Namensgebung „T60-Tabelle“ zu erklären ist (4*5*3=60). Die Bezeichnung TCQ-Profil steht für: Tasktyp, Complexity Class (Komplexitätsklasse) und Service Quality (Dienstgüte).

[6] Abbildung von www.myAMC.de

Abkür-zung	Beschreibung
T	Tasktyp des Dialogschritts (Dialog, Update, Batch, Others)
C	Komplexitätsklasse (1...5)
Q	Service Quality (1 = good, 2 = moderate, 3 = bad)
Cnt	Anzahl der gemessenen Dialogschritte im Messintervall
CPUTi	CPU-Zeit eines Dialogschritts im Workprozess in µsec (= 10^{-6} Sekunden)
DBSU	Maß für die Datenbankaktivität
DBKB	Transferierte Daten in KB vom und zum DB-Server
OthKB	Summe der transferierten Bytes für ABAP/4 Quelltext, ABAP/4-Programm laden und Dynpro / Screen Quelltext, Dynpro / Screen laden
RespTi	Gemessene Antwortzeit am Dispatcher in µsec
DBTi	Die Verweilzeit des Auftrags in der Datenbank (gemessen an der Datenbankschnittstelle)
LockTi	Sperr-Zeiten im R/3-System in µsec (keine DB-Locks)
QueueTi	Wartezeiten am Dispatcher in µsec
*	Zusammenfassung der Daten (Summe bzw. Mittelwert)
Avg....	gemittelter Verbrauch über den Messzeitraum für das entsprechende Attribut

Tabelle 01-1: Attribute eines TCQ-Profils

Die Zeilen der Tabelle entsprechen den verschiedenen Kombinationen aus Tasktyp, Komplexitätsklasse und Dienstgüte. Für die gemessenen Dialogschritte werden zum einen die in der Stunde summierten und zum anderen die gemittelten Verbräuche dargestellt. Eine vollständige Darstellung der T60-Daten enthält weiterhin die Verbräuche für die Tasktypen Batch und Others. Die beschriebene Darstellungsform der TCQ-Profile kann automatisch aus den R/3 Live Monitor-Daten gewonnen werden.

User-Activity-Profil (UA-Tabelle)

Die UA-Tabelle liefert eine nach Mandant, User-Account und Tasktyp gruppierte Sicht auf die gemessenen Dialogschritte. Es werden jeweils die Attribute für die Datenbankaktivität, CPU-Verbräuche und transferierten Datenmengen zwischen Datenbank und Applikationsserver angezeigt. Das User-Activity-Profil bietet somit im Gegensatz zu den TCQ-Profilen die Verknüpfung zwischen gemessenen Dialogschritten und Lastverursachern.

Significant Dialogsteps (Sig-Tabelle)

Im R/3 Live Monitor kann für jede Instanz ein Filter definiert werden, der die Selektion bestimmter Dialogschritte ermöglicht. Die Selektionskriterien basieren auf den Attributen der gemessenen Dialogschritte, wie zum Beispiel:

@ Anteil der Datenbankverweilzeit an der Antwortzeit des Dialogschritts. Nach [6] sollte der Datenbankanteil 50 % nicht übersteigen (für den Dialogbetrieb)

@ Es kann nach Transaktionen (Tcode) gefiltert werden, wie zum Beispiel nach „Z-Transaktionen“ oder basierend auf den Transaktionen nach R/3-Business Components.

Ein produktives R/3-System kann in einer Stunde zum Beispiel 30.000 Dialogschritte verarbeiten. Die zur Betrachtung eines R/3-Systems relevanten Dialogschritte bilden jedoch nur eine kleine Teilmenge aller gemessenen Dialogschritte (höchstens 10 %). Die Filterfunktionalität des R/3 Live Monitors dient dazu, den Umfang der zu betrachtenden Dialogschritte einzugrenzen.

(b) Der WLPSizer im Überblick

Der WLPSizer („Workload Profile Sizer“) ist ein Werkzeug zur Kapazitätsplanung von SAP R/3 Client/Server-Systemen. R/3-Konfigurationen und Workload-Profile können mit Hilfe des WLPSizers in Modelle übertragen und zur automatischen Berechnung von Leistungsmaßen analysiert werden. Typische Leistungsmaße sind hierbei zum Beispiel Antwortzeiten, Durchsätze, Wartezeiten und Auslastungen. Die zur Modellbildung benötigten Daten und die gewonnenen Ausgabeparameter werden in den folgenden Kapiteln dargestellt. Die nachfolgenden Informationen basieren zum größten Teil auf den Angaben des WLPSizer-Handbuchs [3].

Eingabeparameter zur Modellbildung für den WLPSizer

Die Eingabeparameter des WLPSizers beschreiben die Systemkonfiguration und die vom R/3 Live Monitor gemessene Last des R/3-Systems.

Die Basisstruktur eines R/3-Systems besteht aus den vier Grundelementen Clients, Netze, Application-Server und DB-Server. In der Praxis existiert in einem 3-schichtigen R/3-

System eine zentrale Datenbank, die sich auf einem Server befindet (siehe [2]). Der WLPSizer unterstützt die Modellierung von 3-schichtigen und Zentralserver-Systemen. Die vier Grundelemente eines R/3-Systems können im WLPSizer in beliebiger Anzahl[7] eingegeben und zu einem System verbunden werden. Die Spezifikation der einzelnen Systemkomponenten wird durch Bibliotheken unterstützt. Die Bibliotheken enthalten Angaben zu Servern, I/O-Systemen und Netzwerken.

Der WLPSizer kann R/3-Lasten in Form von Workload-Profilen importieren. Als Quelle der Last kann eine Clientgruppe und als Ziel ein Server (Applikations- oder Datenbankserver) ausgewählt werden. Ergänzend zum Workload-Profil wird im WLPSizer der gewünschte oder gemessene Durchsatz (DS/h)[8] mit angegeben. Das Format einer Workload wird in Tabelle 01-2 dargestellt. Die Bezeichnungen entsprechen größtenteils den Angaben der T60-Tabellen des R/3 Live Monitor. Dem WLPSizer werden als Eingabeparameter für das Modell die Daten für CPUTI, DBSU, DBKB und ClBy übergeben. Die zusätzlichen Angaben RespTime, DBTime, LockTime und QueueTime werden für das Modell nicht vorgegeben. Diese Werte können für die Kalibrierung des Modells genutzt werden.

T	C	ratio	CPUTI	DBSU	DBKB	ClBy	RespTime	DBTime	LockTime	QueueTime
D	1	0,5228	19,50	72,84	0,57	325	87,77	20,36	0,14	0
D	2	0,1193	69,46	425,48	11,98	415	421,08	276,03	1,33	0
D	3	0,0322	932,84	1888,59	104,81	603	3959,03	2650,82	6,26	0
D	4	0,0045	6406,56	19454,60	2355,77	715	37959,57	28602,39	12,33	0
D	5	0,0007	73417,97	390184,00	123115,67	1015	379533,32	319135,92	0,11	2
U	1	0,0529	6,54	131,98	0,78	112	20,50	7,31	0,00	0
U	2	0,2076	28,51	595,96	31,14	336	88,35	40,38	0,28	10
U	3	0,0274	99,01	1568,43	191,28	645	676,18	543,27	0,33	6
U	4	0,0018	555,47	6360,15	456,03	812	3455,48	2912,10	0,24	0
U	5	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
B	1	0,0044	12,76	170,98	0,13	225	29,29	10,86	0,31	3
B	2	0,0015	43,20	414,47	20,71	336	607,08	214,62	0,86	2
B	3	0,0011	183,59	1411,25	52,95	548	2286,45	678,01	6,66	1
B	4	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
B	5	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
O	1	0,0143	12,21	59,86	0,00	96	366,25	1,58	0,00	313
O	2	0,0066	29,98	478,31	24,10	223	1606,70	29,55	0,00	8
O	3	0,0028	59,48	1467,97	30,83	469	2072,20	67,20	0,00	864
O	4	0,0001	93,75	5468,00	21,89	712	239,56	178,57	0,00	0
O	5	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
*	*	**1,0000**	**139,59**	**707,05**	**114,79**	**344**	**734,87**	**516,10**	**0,56**	**9**

Tabelle 01-2: Normiertes Workload Profil (Lastprofil) in erweiterter Form

[7] Mit Ausnahme des DB-Servers, der in einem Modell nur einmal existiert.
[8] DS/h = Dialogschritte pro Stunde

Die Beschreibungen für die Abkürzungen der Workload können der folgenden Tabelle entnommen werden:

Abkürzung	Beschreibung
T	Tasktyp des Dialogschritts (Dialog, Update, Batch Others)
C	Komplexitätsklasse (1...5)
ratio	„prozentualer“ Anteil an der gesamten Workload (Anteil / 100)
CPUTI	Mittlere normierte CPU-Zeit eines Dialogschritts im Workprozess
DBSU	Mittlere Anzahl an DB-Service-Units
DBKB	Mittlere Anzahl an Daten-KB, die zwischen der Datenbank und der Applikation transferiert werden
OTHKB	Mittlere Anzahl an zusätzlichen KB
ClBy	Mittlere Anzahl an Bytes, die zwischen Client und Application-Server verschickt werden

Tabelle 01-3: Beschreibung der Workload

Eine Workload beschreibt ein einstündiges Zeitfenster der R/3 Live Monitor-Messung. Es werden für diesen Zeitraum die Messdaten der T60-Tabelle über alle gemessenen Zeitstempel gemittelt und auf einen Dialogschritt normiert. Die Angaben der Workload entsprechen somit den mittleren Verbräuchen pro Dialogschritt.

Die „CPUTI“ entspricht einer mittleren normierten CPU-Zeit eines Dialogschritts im Workprozess. Die Normierung besteht in der Berechnung der CPU-Zeit-Verbräuche für ein Referenzrechner-System, das heißt, es wird berechnet wieviel CPU-Zeit ein Referenzrechner für die gleiche Last benötigt. Die Berechnung basiert auf den in SAPS angegebenen Leistungsdaten der Rechner. Durch diese Normierung wird die Grundlage geschaffen, dass Workloads für die Modellierung alternativer R/3-Systeme wiederverwendbar werden. Die Angaben der „DBSU“, „DBKB“ und „Clientbytes“ entsprechen den Daten der T60-Tabelle. Die Workload-Profile können automatisch aus den R/3 Live Monitor-Daten gewonnen werden.

Ausgabeparameter des WLPSizers

Nach Beendigung der Eingabe des R/3-Systems und der R/3-Last transformiert der WLPSizer für den Anwender unsichtbar die Eingangsdaten in eine analytisch behandelbare

Darstellung, welche in wenigen Sekunden gelöst wird (siehe [4], [5]). Dem Anwender werden verschiedene Sichten auf die Ergebnisse geboten:

- *Server-View:* Darstellung der Ergebenisse reduziert auf einen bestimmten Server, d.h. es werden die von diesem Server verarbeiteten Aufträge angezeigt.
- *Workload-View:* Darstellung der Ergebnisse reduziert auf eine bestimmte Workload, d.h. es werden die Verbräuche dieser Workload im System dargestellt.
- *Utilization-View:* Darstellung der prozentualen Auslastungen der einzelnen Systemkomponenten reduziert auf eine bestimmte Workload.
- *I/O-View:* Darstellung der I/O-Auslastung.

Mit Ausnahme der I/O-View werden die Sichten als tabellarische Übersicht über die verarbeiteten Dialogschritte im R/3-System dargestellt. In Anlehnung an die Eingabedaten werden auch im Ergebnis die Dialogschritte nach Tasktypen und Komplexitätsklassen gruppiert. In den verschiedenen Sichten werden für die Dialogschritte die berechneten Leistungsmaße, wie zum Beispiel Durchsätze, Antwortzeiten und Verweilzeiten angezeigt.

Es existiert weiterhin die Möglichkeit "Service Levels" zu definieren, die, basierend auf der Response Time und differenziert nach Komplexitätsklasse und Tasktyp des Dialogschritts, definiert werden können. Die Güteklassen der Service Levels entsprechen "good", "moderate" und "bad". Der R/3 Live Monitor definiert die Service-Güte-Klassen in Abhängigkeit zum DBSU-Verbrauch eines einzelnen Dialogschritts, wobei der WLPSizer für die unterschiedlichen Tasktypen und Komplexitätsklassen Schwellwerte bez. der Antwortzeit der Dialogschritte definiert.

Prognosen für Lastzuwächse, Hardware-Upgrades und R/3-Neusysteme

Prognosen werden hauptsächlich auf Grund von Änderungen und Erweiterungen der Geschäftsprozesse benötigt. Die notwendigen Veränderungen können im Hardware-, zum Beispiel Aufrüstung der Server mit weiteren CPUs, sowie im Software-Bereich, R/3-Release-Wechsel, erfolgen. Eine der Hauptfragen, die sich in diesem Kontext stellt, ist: „Welche Auswirkungen hat die Veränderung auf die Performance des Systems und die Dienstgüte der zu verarbeitenden Aufträge?"

Es muss die Frage beantwortet werden, ob das bestehende System mit den Neuerungen die benötigten Anforderungen bez. Durchsatz und Antwortzeit erbringen kann. Zur Beantwortung dieser Frage und Gewinnung von Prognosen wird mit Hilfe des R/3 Live Monitor (myAMC.LNI) die Last im R/3-System zu Hochlastzeiten gemessen. Basierend auf diesen Messdaten folgt die Erstellung und anschließende Kalibrierung eines Basismodells des

existierenden Systems mit dem WLPSizer. Das Basismodell stellt nun die Ausgangssituation für die folgenden Prognosen dar.

Prognosen für einen R/3-Release-Wechsel

Der Release-Wechsel wird im Modell mit Hilfe der von SAP veröffentlichten prozentualen Lastzuwächse realisiert. Zum Beispiel wird der Wechsel von Release 4.0B auf 4.6 mit einem Lastzuwachs von 35 % angegeben. Es besteht somit die Möglichkeit die Leistungskennzahlen aller Server der Konfiguration um 35 % zu verringern, so dass die Server reduzierte SAPS-Zahlen in der WLPSizer-Konfiguration zugeordnet bekommen. Es ist somit ein neues (Prognose-)Modell entstanden. Als Ergebnis des neuen Modells werden Durchsätze und Antwortzeiten des R/3-Systems zu Hochlastzeiten unter Einfluss des neuen R/3-Release 4.6 geliefert. Die Ergebnisse können somit eine erste Prognose liefern, wie sich das R/3-System mit neuem Release verhält und ob die vorhandene Hardware weiterhin ausreicht. Es können bereits durch das Modell durch den Release-Wechsel verursachte mögliche Engpässe erkannt werden.

Prognosen für ein Hardware-Upgrade

Die Modellierung eines Hardware-Upgrades ist mit einem zugrunde liegenden Basismodell relativ schnell realisiert. Mit Hilfe der Bibliotheken des WLPSizers können die bestehenden Systemkomponenten gegen leistungsstärkere Komponenten ersetzt werden. Das Ergebnis des Modells kann Aufschluss über die nötige Dimensionierung der zu ersetzenden Hardware geben und eventuell auch Seiteneffekte aufdecken, die ein Hardware-Upgrade verursachen könnte. Weiterhin ist es möglich, dass das Modell Alternativen für ein Hardware-Upgrade bietet, wie zum Beispiel, dass nicht der Datenbank-Server mit Prozessoren erweitert werden muss, sondern dass die Verbesserung des I/O-Systems bereits eine Lösung des Performance-Problems darstellt.

Prognosen für ein R/3-Neusystem

Für die Planung eines R/3-Neusystems kann begleitend zum durchgeführten Sizing eine Modellierung auf Basis der Sizing-Daten durchgeführt werden. Ein Sizing liefert die folgenden Ergebnisse:

- Anwender-Daten-Sicht: Es werden in Abhängigkeit der aktiven User, der Denkzeit, der Modi und eines kundenspezifischen Lastfaktors die SAPS-Werte für Dialog, Update und der Datenbank der einzelnen R/3-Business Components berechnet.
- Transaktions-Daten-Sicht: Für einzelne Transaktionen werden in Abhängigkeit von Durchsatz (Anzahl/Stunde), Belegen und kundenspezifischem Lastfaktor SAPS-Werte für Dialog, Update und Datenbank berechnet.

Die für die Modellierung benötigten Workloads werden auf Basis der im Sizing errechneten SAPS-Werte erstellt.

01.05 Zusammenfassung

Das beschriebene Vorgehen und die Werkzeuge zum R/3-Kapazitätsmanagement sind allgemein einsetzbar und haben sich in vielen Projekten bewährt. Der Vorteil der Kunden besteht in einer bestmöglichen Verlässlichkeit der Prognose und in der Möglichkeit des unmittelbaren Vergleichs zwischen erwartetem und tatsächlichem Lastaufkommen und dem zugehörigen Leistungsverhalten. Durch den Beitrag des Kapazitätsmanagements zur Qualitätssicherung in der Realisierung und Integration können „böse Überraschungen“ bei Produktionsbeginn vermieden werden.

01.06 Literatur und Referenzen

[1] Fujitsu Siemens Computers: "Handbuch zum R/3 Live Monitor Version 4.0"

[2] Rüdiger Buck-Emden: "Die Technologie des SAP R/3 Systems", 1998

[3] Universität Essen, Institut für Informatik: "WLPSizer 1.5 Handbuch", 2001

[4] D. A. Menascé, V. A. F. Almeida, L. W. Dowdy: "Capacity Planning and Performance Modelling: From Mainframes to Client-Server Systems", Prentice-Hall PTR, Englewood Cliffs, 1994

[5] D. A. Menascé, V. A. F. Almeida: "Capacity Planning for Web Performance", Prentice-Hall PTR, Englewood Cliffs, 1998.

[6] Thomas Schneider: "SAP R/3 Performanceoptimierung", Addison Wesley, 1999

KAPITEL 02 MONITORING UND BENCHMARKING FÜR DAS R/3-PERFORMANCE-ENGINEERING

KAY WILHELM, JÜRGEN PFISTER,
CHRISTIAN KOWARSCHICK, STEFAN GRADEK

02.01 Einleitung

Die SAP AG ist Marktführer im Bereich betrieblicher Anwendungssoftware. Ihr Erfolg in den 90er Jahren gründet sich auf das klassische ERP[9] Produkt SAP R/3. Mit mySAP.com verlagert die SAP mit der Jahrtausendwende den Schwerpunkt ihrer Geschäftstätigkeit ins Internet und bietet umfassende Softwarelösungen für die integrierte überbetriebliche Zusammenarbeit im Internet an. mySAP.com umfasst das vollständige Softwareportfolio der SAP einschließlich der Lösungen für Personalwirtschaft, Rechnungswesen, Customer Relationship Management, Supply Chain Management, Business Intelligence und Product Lifecycle Management.

Die in einer Firma definierten Geschäftsprozesse werden mittels mySAP.com realisiert und verarbeitet. Es muss gewährleistet werden, dass das zugrunde liegende System ausreichend Ressourcen für die Verarbeitung der Geschäftsprozesse bereithält (Sizing) und diese hinsichtlich der Antwortzeit in vorgegebener Zeit (Service Level) verarbeiten kann. Können diese Zusicherungen nicht eingehalten werden, kann es zu erhöhten Verzögerungen in der Verarbeitung der Geschäftsprozesse oder sogar zu Systemausfällen kommen. In Abhängigkeit des Einsatzgebiets von SAP können somit hohe Verluste für die Firma entstehen. Das SAP-Performance-Engineering beschäftigt sich mit der Analyse von SAP-Systemen. Ziel ist hierbei die Verhinderung von System-Engpässen und Ausfällen mittels einer den SAP-Betrieb begleitenden Langzeitüberwachung des Systems. In diesem Beitrag werden Grundlagen für das SAP-Performance-Engineering beschrieben, und zwar die Langzeitüberwachung durch Monitoring und das Benchmarking. Aufgrund langjähriger Erfahrungen steht das klassische R/3 im Mittelpunkt, welches nach wie vor eine zentrale Komponente innerhalb der mySAP.com Strategie darstellt.

Nach einer kurzen Beschreibung von R/3 im zweiten Abschnitt, werden danach zum einen die Methoden und Techniken für die Überwachung und Analyse eines R/3-Systems (Ab-

[9] ERP = Enterprise Resource Planning

schnitt 02.3) und zum anderen die Werkzeuge (Abschnitt 02.4) beschrieben. Abschnitt 02.5 enthält Hintergrundinformationen zum Thema SAP Standard Application Benchmarks. Den Abschluss bilden die in Abschnitt 02.6 beschriebenen Beispiele zur Anwendung der vorgestellten Methoden und Werkzeuge.

02.02 SAP R/3 – Ein Standard-Software-Paket

SAP R/3 ist eine Software, welche Lösungen zur Unterstützung und Vervielfachung der Geschäftprozesse eines Unternehmens anbietet. R/3 enthält dazu sog. Business Components (Module), die bestimmte Funktionen ausführen, wie zum Beispiel:

- Produktionsplanung (PP)
- Vertrieb (SD)
- Kontrolle und Steuerung (CO)
- Personalwirtschaft (HR)
- Finanzwesen (FI)
- ... und andere

Weiterhin stehen Anwendungen und Werkzeuge zur Eigenentwicklung und Anpassung an individuelle Gegebenheiten (Customizing) zur Verfügung. Das Computer Center Management System (CCMS) sammelt in einem R/3-System Performance-Daten und stellt Funktionen zur System-Analyse bereit.

SAP R/3 unterstützt durch seine offene Realisierung eine Vielzahl von Betriebssystemen (Windows NT, Windows 2000, Unix, Solaris etc.), Datenbanken (Oracle, Informaix, Sybase etc.) und Front End Präsentationssoftware (Windows, OS/2, OSF/Motif etc.).

(a) Architektur von SAP R/3-Systemen

Ein SAP R/3-System kann in drei Schichten unterteilt werden:

- **Präsentation:** Darstellung der Daten und Benutzer-Interaktion
- **Applikation:** Bereitstellung der Anwendungslogik
- **Datenhaltung:** Datenverwaltung mittels eines Relationalen Datenbank Management Systems (RDBMS).

Die drei Schichten können durch verschiedene Konfigurationen realisiert werden (siehe Abbildung 02-1, vgl. [1]).

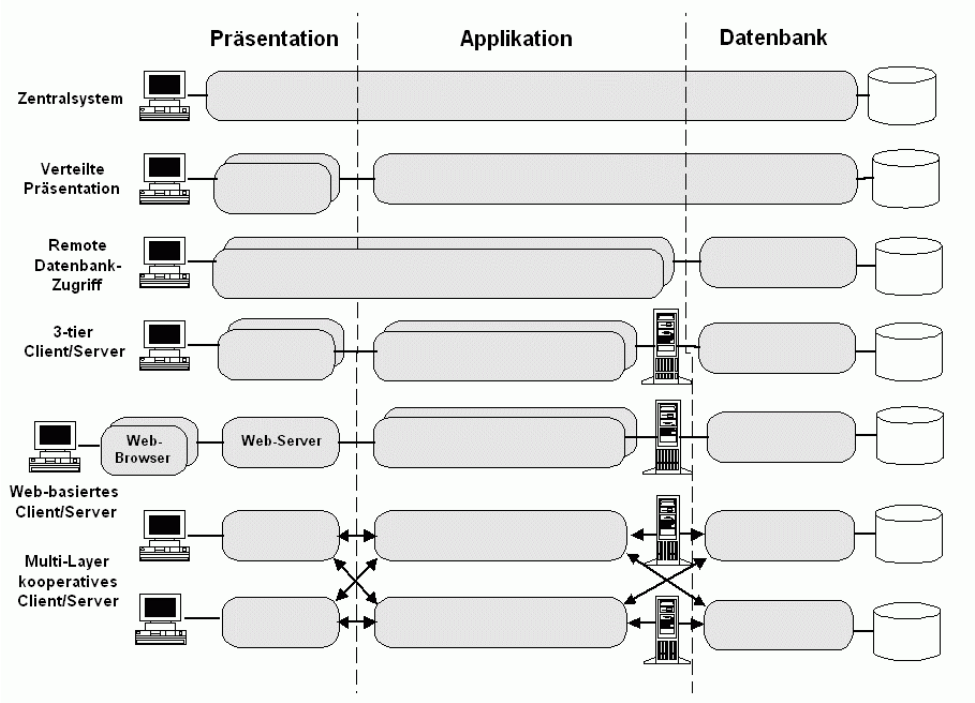


Abbildung 02-1: R/3-Konfigurationen

Im Zentralsystem (Zentralserversystem) befinden sich alle Ebenen auf einem Rechner, was eine beschränkte Skalierbarkeit zur Folge hat. Bei einer zweistufigen Architektur sind die Schichten auf zwei Rechnertypen verteilt. Dabei können entweder Präsentationsserver eingerichtet werden und Applikation und Datenbank befinden sich auf einem Rechner, oder es können Präsentation und Applikation auf gemeinsamen Rechnern realisiert sein und es gibt einen Datenbankserver. Bei einer dreistufigen Architektur übernimmt jede Hardwareplattform eine spezielle Funktionalität, das heißt jeder Rechner ist genau einer Schicht zugeordnet. Dies ermöglicht eine Auswahl optimaler Hardware für das spezielle Aufgabengebiet. Vierstufige Konfigurationen haben sich gemeinsam mit dem Internet-Dienst WWW (World Wide Web) auf dem Markt etabliert. Zusätzlich zu den bisherigen Diensten existiert ein Web-Server zur Bereitstellung der gewünschten Seiten. Bei einer mehrstufigen, kooperativen Architektur werden unterschiedliche Anwendungs- und Systemdienste auf die für die jeweilige Aufgabe am besten geeigneten Rechner verteilt.

Im Wesentlichen besteht ein R/3-System somit aus den vier Grundelementen: Clients (bzw. Präsentationsserver), Netze, Application-Server und DB-Server.

02.03 Messkonzept für R/3-Last

SAP R/3 ist eine transaktionsorientierte Anwendung. Alle durchgeführten Transaktionen werden vom R/3-System protokolliert. Die Transaktionen werden mit Hilfe der vom Dispatcher verwalteten Workprozesse abgearbeitet, die in vier Tasktypen eingeteilt werden:

- @ Dialog (Dialogverarbeitung)
- @ Update (Verbuchungen)
- @ Batch (Hintergrundverarbeitung)
- @ Others (Spool, Enqeue etc.)

Der Dialog-Anteil einer Transaktion wird in Dialogschritte[10] unterteilt. Jeder Dialogschritt wird durch einen Dialog-Workprozess bearbeitet. Die Verbuchungen einer Transaktion werden in Verbuchungskomponenten aufgeteilt, die den Verbuchungs-Workprozessen zugewiesen werden. Batch-Jobs werden den Background-Workprozessen zur Verarbeitung übergeben. Im Folgenden wird jegliche Durchführung eines Workprozesses als Dialogschritt[11] bezeichnet. Das R/3-System verwaltet eine Liste aller Dialogschritte, die im R/3-System ausgeführt wurden (Statistical-Records). Für jeden Dialogschritt werden zum Beispiel System, Instanz, Antwortzeit, CPU-Zeit, Datenbank-Aktivität, Transaktions-Code, Benutzer-ID und Report-Name erfasst. Insgesamt existieren pro Dialogschritt ca. 80 Attribute. Die Anzahl der Attribute kann in Abhängigkeit des R/3-Release variieren, da mit steigendem Release die Dialogschritt-Eigenschaften zum Teil ergänzt werden, wie zum Beispiel unter Release 4.6c die Client-Zeiten und Verbräuche neu hinzugekommen sind.

Die Bewertung der ermittelten Performance-Daten erfolgt im Workload-Monitor des CCMS anhand von Mittelwerten über alle gemessenen Dialogschritte oder mittels eines Tasktyp-Profils gruppiert nach Tasktypen. Es wird hierbei nicht unterschieden, ob zum Beispiel der gemessene Dialogschritt einen Zeilenwechsel oder eine komplexe Such-Anfrage repräsentiert. Da in einem R/3-System mittels Mausklicks, Menünavigation, Zeilenumbrüchen etc. eine hohe Anzahl von Dialogschritten mit sehr gutem Zeitverhalten erzeugt werden kann, ist der globale Mittelwert als alleiniges Bewertungskriterium unbrauchbar. Es wird stattdessen eine von der Komplexität der Dialogschritte abhängige Bewertung bez. Ressourcenverbrauch und Antwortzeit benötigt.

---

[10] Ein Dialogschritt entspricht der Verarbeitung eines Bildschirms bzw. einer Eingabemaske.

[11] Diese Sprachregelung wurde von der SAP geprägt. Es besteht somit eine atomare Einheit über alle Tasktypen.

(a) Komplexitätsklassen

Es wird ein skalares Komplexitätsmaß für Dialogschritte eingeführt, das von der verwendeten Hardware-Plattform, dem R/3-Release und der verwendeten Systemsoftware unabhängig ist. Als Maß für die Komplexität eines Dialogschrittes werden DB-Service-Units (DBSU) verwendet, die, basierend auf der Anzahl der bearbeiteten Sätze (Records Transferred) und den logischen und physikalischen Datenbankaufrufen (Insert, Delete, Update, Read Sequential, Read Direct und Commit), als gewichtete Summe berechnet werden. Die Dialogschritte werden in fünf Komplexitätsklassen unterteilt, wobei Komplexitätsklasse 1 die einfachen Dialogschritte mit geringer Datenbankaktivität (<= 200 DBSU) und Komplexitätsklasse 5 die großen Dialogschritte mit einer hohen Datenbankaktivität (> 100 000 DBSU) enthält. Die Schwellwerte für die Einteilung der Komplexitätsklassen können der folgenden Tabelle entnommen werden.

Komplexitätsklasse	Beschreibung	DBSU
1	Small simple actions	<= 200
2	Simple actions	<= 800
3	Middle size actions	<= 5000
4	Complex actions	<= 100 000
5	Very complex actions	> 100 000

Tabelle 02-1: Komplexitätsklassen

Definiert und validiert wurden die Schwellwerte anhand von SD-Benchmarks[12] (SD = Sales and Distribution).

(b) Dienstgüteklassen

Die Dienstgüteklassen stellen eine Bewertung der Antwortzeit im Verhältnis zur Komplexität des Dialogschritts dar. Es werden die Güteattribute good, moderate und bad vergeben. Die Definition der Dienstgüteklassen ist in der folgenden Tabelle dargestellt.

[12] Durchführung der Benchmarks von Fujitsu Siemens Computers.

Dienst-güteklasse	Kompl.[13] Klasse 1	Kompl. Klasse 2	Kompl. Klasse 3	Kompl. Klasse 4	Kompl. Klasse 5
Good	<= 200	<= 1000	<= (#DBSU)[14]	<= (#DBSU)	<= (#DBSU)
Moderate	<= 800	<= 4000	<= 4*(#DBSU)	<= 4*(#DBSU)	<= 4*(#DBSU)
Bad	> 800	> 4000	> 4*(#DBSU)	> 4*(#DBSU)	> 4*(#DBSU)

Tabelle 02-2: Dienstgüte- und Komplexitätsklassen (alle Angaben in Millisekunden)

Es ist hierbei zu beachten, dass für die Komplexitätsklassen 1 und 2 feste Schranken für die Antwortzeit vorgegeben werden und ab Komplexitätsklasse 3 die Antwortzeit in Abhängigkeit der tatsächlich verbrauchten DBSU bewertet wird. So wird zum Beispiel die Antwortzeit für einen Dialogschritt der Komplexitätsklasse 4 verglichen mit den von ihm verbrauchten DBSU (#DBSU)ms.

[13] Kompl. = Komplexitätsklasse
[14] (#DBSU)ms = 1 ms pro DB-Service-Unit

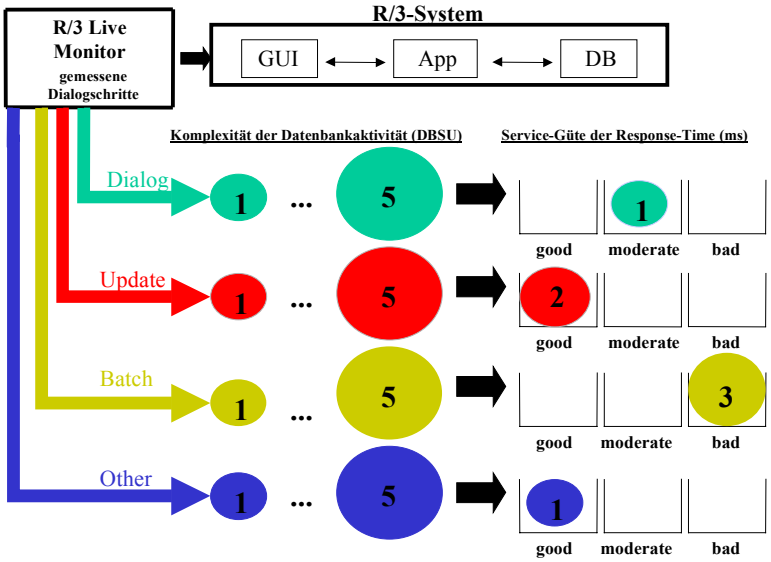


Abbildung 02-2: Unterscheidung in Tasktyp, Komplexitäts- und Dienstgüteklasse

Die Bezeichnung #DBSU in Tabelle 02-2 bedeutet, dass man von einer Zuordnung 1 ms pro DBSU ausgeht. Somit gilt, dass zum Beispiel 6000 DBSU genau 6000 ms entsprechen. Liegt die Antwortzeit unter (#DBSU)ms, so entspricht der Dialogschritt der Dienstgüte *good*. Bei einer Antwortzeit unter 4*(#DBSU)ms entspricht der Dialogschritt der Dienstgüte *moderate*, ansonsten muss der Dialogschritt der Dienstgüte *bad* zugeordnet werden. Zusammenfassend kann man die Einteilung der Dialogschritte in Tasktyp, Komplexitätsklasse und Dienstgüte anhand der Abbildung 02-2 veranschaulichen.

02.04 Monitoring-Werkzeuge

In diesem Abschnitt werden die zur Überwachung eines SAP R/3-Systems benötigten Werkzeuge beschrieben. Es wird zum einen das Messwerkzeug R/3 Live Monitor (myAMC.LNI) beschrieben, das die Last im R/3-System misst und hierzu die bereits beschriebene Messmethodik anwendet. Weiterhin werden Programme vorgestellt, die zur Messung auf Betriebssystem- und Datenbank-Ebene eingesetzt werden können.

(a) R/3 Live Monitor (myAMC.LNI)

Das von Siemens entwickelte Werkzeug „R/3 Live Monitor“ (myAMC.LNI) ist ein Enterprise IT Management Tool für SAP R/3-Systeme. Es können mit diesem Werkzeug verschiedene SAP R/3-Systeme von einer zentralen Stelle aus überwacht werden (Single Point of R/3 Control). Das Werkzeug besitzt eine Basis- und Expert-Management-Variante. Die folgenden Eigenschaften stellen die wesentlichen Funktionen und Eigenschaften des Basis-Managements dar:

- Übersichtliche Darstellung der Verteilung sowie der Beziehungen und Aktivitäten von R/3-Datenbanken und ihrer zugehörigen R/3-Applikations-Instanzen
- Erkennung und Zuordnung von Performanceproblemen in einem R/3-System
- Anzeige von SAP R/3-Alerts (Alarmautomatik)

Eine Erweiterung stellt das „R/3 Live Monitor Expert-Management“ dar, das zusätzlich zu den bisherigen Basis-Management-Eigenschaften die Komponente „LNISCA[15]“ beinhaltet. Die Erweiterung des R/3 Live Monitors liegt in der Analyse der in einem SAP R/3-System anfallenden Messdaten, die das System standardmäßig für jede Transaktion in Form von Dialogschritten protokolliert. Der LNISCA besitzt die folgenden Hauptaufgaben:

- R/3 Workload Management
- R/3 Database Service
- R/3 Service Quality
- User Activity Measurement
- Tracing Significant Transactions
- Accounting Management
- Exception Logging

Zur Darstellung der gemessenen Dialogschritte bietet der R/3 Live Monitor verschiedene Sichten auf die Daten, die in der folgenden Grafik dargestellt werden.

[15] SCA = Service Quality, Capacity and Accounting

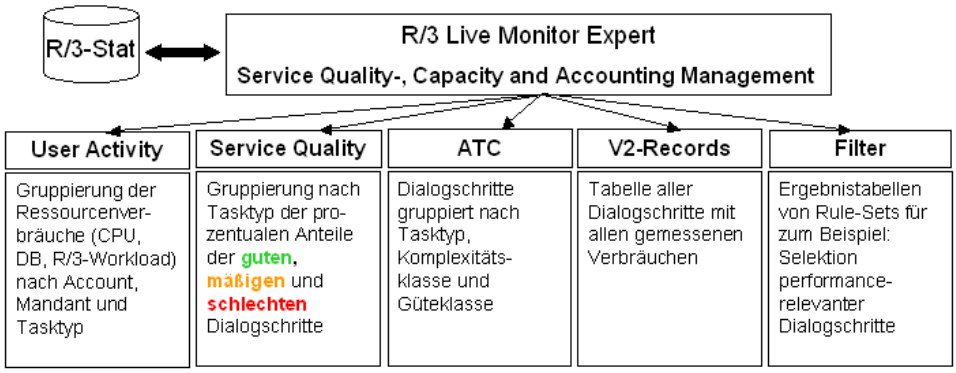

R/3-Stat
R/3 Live Monitor Expert
Service Quality-, Capacity and Accounting Management
User Activity
Service Quality
ATC
V2-Records
Filter
Gruppierung der Ressourcenver-bräuche (CPU, DB, R/3-Workload) nach Account, Mandant und Tasktyp
Gruppierung nach Tasktyp der pro-zentualen Anteile der guten, mäßigen und schlechten Dialogschritte
Dialogschritte gruppiert nach Tasktyp, Komplexitäts-klasse und Güteklasse
Tabelle aller Dialogschritte mit allen gemessenen Verbräuchen
Ergebnistabellen von Rule-Sets für zum Beispiel: Selektion performance-relevanter Dialogschritte

Abbildung 02-3: Ausgaben des R/3 Live Monitor

In den folgenden Abschnitten werden die TCQ-Profile, User-Activity-Profile und Significant Dialogsteps als die drei wichtigsten Darstellungsformen für die R/3-Kapazitätsplanung beschrieben.

TCQ-Profile (T60-Tabelle)

In einem R/3-System[16] werden in der Regel mehrere Tausend bis hin zu mehrere Hunderttausend Dialogschritte verarbeitet. Zur Reduktion der Datenmenge werden die gemessenen Dialogschritte zu sogenannten TCQ-Profilen aggregiert, deren Zeilen keine einzelnen Dialogschritte, sondern eine Gruppe der in einem bestimmten Zeitzyklus angefallenen Dialogschritte repräsentieren. Die Attribute der T60-Tabelle entsprechen den aufsummierten Verbräuchen aller Dialogschritte einer Gruppe. Die Dialogschritte werden ergänzend zu den vier Tasktypen in fünf Komplexitätsklassen und drei Dienstgüteklassen unterteilt, wodurch auch die Namensgebung „T60-Tabelle“ zu erklären ist (4*5*3=60). Die Bezeichnung TCQ-Profil steht für: Tasktyp, Complexity Class (Komplexitätsklasse) und Service Quality (Dienstgüte). Die nachfolgende Tabelle entspricht in leicht veränderter Form einem Auszug eines TCQ-Profils. Die Daten repräsentieren die in einer Stunde gemessenen Dialogschritte (Tausendertrennzeichen = Dezimalpunkt).

[16] Es wird hierbei von einer Standard-R/3-Konfiguration ausgegangen, wie sie bei einem mittelständischem Unternehmen vorzufinden ist.

T	C	Q	Cnt	CPUTi [ms]	DBSU [count]	DBKB [kb]	OthKB [kb]	RespTi [ms]	DBTi [ms]	LockTi [ms]	QueueTi [ms]
D	1	1	5.532	76.047	389.083	2.614	2	122.009	37.483	356	1.264
D	1	2	239	18.938	29.343	624	0	87.379	39.577	420	61
D	1	3	65	18.813	6.695	147	0	302.827	41.744	13	15
D	2	1	1.189	68.812	496.230	12.413	1	268.638	180.238	1.477	374
..	..	..	..	..	..	..	..	..	..	..	..
U	1	1	586	3.813	77.282	443	0	7.777	4.271	2	141
U	1	2	1	16	131	0	0	597	11	0	0
U	1	3	3	31	456	31	0	3.722	31	0	0
U	2	1	2.292	64.844	1.364.104	72.627	0	149.949	85.750	649	1.889
..	..	..	..	..	..	..	..	..	..	..	..
*	*	*	11.162	1.569.875	7.958.236	1.334.910	5.286	8.202.635	5.760.708	6.299	103.803
D	*	*	7.585	1.448.875	5.764.069	1.187.431	5.091	7.428.636	5.421.329	5.422	1.824
U	*	*	3.233	111.328	2.065.860	143.634	18	492.833	322.361	767	24.361
B	*	*	78	3.563	32.359	1.018	177	39.193	12.317	110	170
O	*	*	266	6.109	95.948	2.828	0	241.973	4.701	0	77.448

T	C	Q	Cnt	AvgCPU [ms]	AvgDBSU [count]	AvgDBKB [kb]	AvgRespTi [ms]	AvgDBTi [ms]
D	1	1	5.532	14	70	0	22	7
D	1	2	239	79	123	3	366	166
D	1	3	65	289	103	2	4.659	642
D	2	1	1.189	58	417	10	226	152
..	..	..	..	..	..	..	..	..
U	1	1	586	7	132	1	13	7
U	1	2	1	16	131	0	597	11
U	1	3	3	10	152	10	1.241	10
U	2	1	2.292	28	595	32	65	37
..	..	..	..	..	..	..	..	..
*	*	*	11.162	141	713	120	735	516
D	*	*	7.585	191	760	157	979	715
U	*	*	3.233	34	639	44	152	100
B	*	*	78	46	415	13	502	158
O	*	*	266	23	361	11	910	18

Tabelle 02-3: TCQ-Profil

Die Attribute eines TCQ-Profils sind in der folgenden Tabelle beschrieben.

Abkürzung	**Beschreibung**
T	Tasktyp des Dialogschritts (Dialog, Update, Batch, Others)
C	Komplexitätsklasse (1...5)
Q	Dienstgüte (1 = good, 2 = moderate, 3 = bad)
Cnt	Anzahl der gemessenen Dialogschritte im Messintervall
CPUTi	CPU-Zeit eines Dialogschritts im Workprozess in µsec (= 10^{-6} Sekunden)
DBSU	Maß für die Datenbankaktivität
DBKB	Transferierte Daten in KB zwischen DB-Schnittstelle der Appl.-Server und DB-Server
OthKB	Summe der transferierten Bytes für ABAP/4 Quelltext, ABAP/4-Programm laden und Dynpro / Screen Quelltext, Dynpro / Screen laden
RespTi	Gemessene Antwortzeit am Dispatcher in µsec
DBTi	Die Verweilzeit des Auftrags in der Datenbank (gemessen an der Datenbankschnittstelle)
LockTi	Zeiten für R/3-Locks in µsec (keine DB-Locks)
QueueTi	Wartezeiten am Dispatcher in µsec
*	Zusammenfassung der Daten (Summe bzw. Mittelwert)
Avg....	Gemittelter Verbrauch über den Messzeitraum für das entsprechende Attribut

Tabelle 02-4: Attribute eines TCQ-Profils

Die Zeilen von Tabelle 02-3 entsprechen den verschiedenen Kombinationen aus Tasktyp, Komplexitätsklasse und Dienstgüte. Für die gemessenen Dialogschritte werden zum einen die in der Stunde summierten und zum anderen die gemittelten Verbräuche dargestellt. Eine vollständige Darstellung der T60-Daten enthält weiterhin die Verbräuche für die Tasktypen Batch und Others. Die beschriebene Darstellungsform der TCQ-Profile wird automatisch vom R/3 Live Monitor aus den R/3-Daten generiert. TCQ-Profile können in einem R/3-System die Last verschiedener Objekte beschreiben, wie zum Beispiel einer Instanz, eines Moduls, einer Transaktion oder auch eines Geschäftsprozesses. Es könnte somit zum Beispiel für jedes R/3-Modul jeweils ein TCQ-Profil erstellt werden.

User-Activity-Profil (UA-Tabelle)

Die UA-Tabelle liefert eine nach Mandant, User-Account und Tasktyp gruppierte Sicht auf die gemessenen Dialogschritte. Es werden jeweils die Attribute für die Datenbankaktivität (DBSU), CPU-Verbräuche (CPUTi) und transferierten Datenmengen zwischen Datenbank und Applikations-Server (DBKB) angezeigt. Das User-Activity-Profil bietet somit im Gegensatz zu den TCQ-Profilen die Verknüpfung zwischen gemessenen Dialogschritten und Lastverursachern.

Significant Dialogsteps (V2Sig-Tabelle)

Im R/3-Live Monitor kann für jede Instanz ein Filter definiert werden, der die Selektion bestimmter Dialogschritte ermöglicht. Die Selektionskriterien basieren auf den Attributen der gemessenen Dialogschritte, wie zum Beispiel:

@ Der Anteil der Datenbankzeit (DBTi) an der Antwortzeit des Dialogschritts (RespTi) sollte nach [2] 50 % nicht übersteigen (für den Dialogbetrieb).

@ Es kann nach Transaktionen (Tcode) gefiltert werden, wie zum Beispiel nach „Z-Transaktionen“ oder basierend auf den Transaktionen nach R/3-Business Components.

Ein produktives R/3-System kann in einer Stunde über 100.000 Dialogschritte verarbeiten. Die zur Betrachtung eines R/3-Systems relevanten Dialogschritte bilden jedoch nur eine kleine Teilmenge aller gemessenen Dialogschritte (höchstens 10 %). Die Filterfunktionalität des R/3 Live Monitors dient dazu, den Umfang der zu betrachtenden Dialogschritte einzugrenzen.

(b) Windows NT Systemmonitor

Microsoft Windows NT bietet bereits nach der Installation viele Möglichkeiten der Performance-Überwachung. Es können über die Performance Registry[17] Daten gesammelt und mit Hilfe des Windows NT Systemmonitors ([3], [4]) dargestellt werden. Die Performance-Daten lassen sich in Basisinformationen und erweiterte Informationen unterteilen. Die Basisinformationen beschreiben:

[17] Abschnitt der Registry von Windows NT für die Performance-Daten (siehe [4]). Unter Registry versteht man die Systemregistrierung, die Informationen über den Computer enthält.

- Prozessoren
- Speicher
- I/O
- Netzwerk

Die erweiterten Informationen werden durch applikationsspezifische Counter[18], wie zum Beispiel MS SQL Server oder MS Exchange Server, geliefert.

In Windows NT werden logisch zusammenhängende Performance-Daten als Objekte zusammengefasst. Jedem Objekt werden wiederum Counter zugewiesen, die die statistischen Datenfelder der Performance Registry repräsentieren. Eine „Performance Monitoring"-Schnittstelle bietet die Möglichkeit, die Daten aus der Performance Registry abzufragen, so dass zum Beispiel der Windows NT Systemmonitor die Daten lesen und darstellen kann. Der Systemmonitor entspricht somit einer „Überwachungszentrale von Windows NT für die Ressourcennutzung" [3, Seite 773]. Ergänzend zu den Basis-Performance-Werten können für jede NT-Applikation eigene Performance-Counter definiert und in der Performance Registry eingetragen werden. Vom Systemmonitor gemessene Daten können grafisch als Kurven über den gesamten Messzeitraum (bzw. Teilausschnitte) angezeigt und als „Comma Separated Values" (CSV) exportiert werden. Die exportierten Dateien können anschließend mit Hilfe von MS Excel weiterverarbeitet werden.

Die mit Hilfe des Windows NT Systemmonitors gemessenen Daten liefern einen wesentlichen Beitrag zum Verständnis der IST-Situation des zu betrachtenden SAP R/3-Systems. Es können zum Beispiel bereits vorhandene „Bottlenecks" einzelner Systemkomponenten und Hochlastphasen entdeckt werden. Die Tabelle 02-5 enthält eine Auswahl der Basis-Performance-Daten von Windows NT, die zur Analyse eines Systems herangezogen werden können.

[18] Counter = In der Registry definierte Variablen zur Speicherung von Performance-Werten.

Objects / Counter	Bedeutung
CPU / CPU-Time	CPU-Auslastung für jede einzelne CPU
Process bzw. Thread	Anteile von SAP bzw. Datenbank an der Gesamtlast
Memory / Available Bytes	Zur Verfügung stehender freier Hauptspeicher
Memory / Page Reads	Paging-Aktivität mit Zugriff auf das I/O-System
File Cache / System Cache Resident Bytes	Größe des File Cache; maximal 512 MB sind unter NT möglich; dieser Cache sollte auf einem R/3-System nur 7 %-10 % des physischen Speichers einnehmen (siehe [2])
File Cache / Cache Hit Ratios	Cache Hit Ratios für die unterschiedlichen Zugriffsarten (Lazy Write, Copy-Interface, Mapping, MDL, Write Through)
I/O-System / Avg. Disk Queue Length	Betrachtung der IO-Auslastung anhand der wartenden IO-Prozesse
I/O-System / Disk Read Bytes/sec I/O-System / Disk Write Bytes/sec	Transferiertes Datenvolumen in Bytes
Prozesse / Working-Set	Speicherbelegung des realen Speichers durch aktive Prozesse (Vorsicht: dieser Wert ist durch Shared DLLs immer größer als der tatsächlich verwendete Speicher)

Tabelle 02-5: Windows NT Systemmonitor-Daten

Die folgenden Abbildungen stellen Beispiele für Windows NT Systemmonitor-Daten dar. Das erste Diagramm zeigt Auslastungen pro Prozessor sowie die CPU-Mittelwertkurve in einem Zeitraster von 1 Minute von 15.00 Uhr bis 16.00 Uhr.

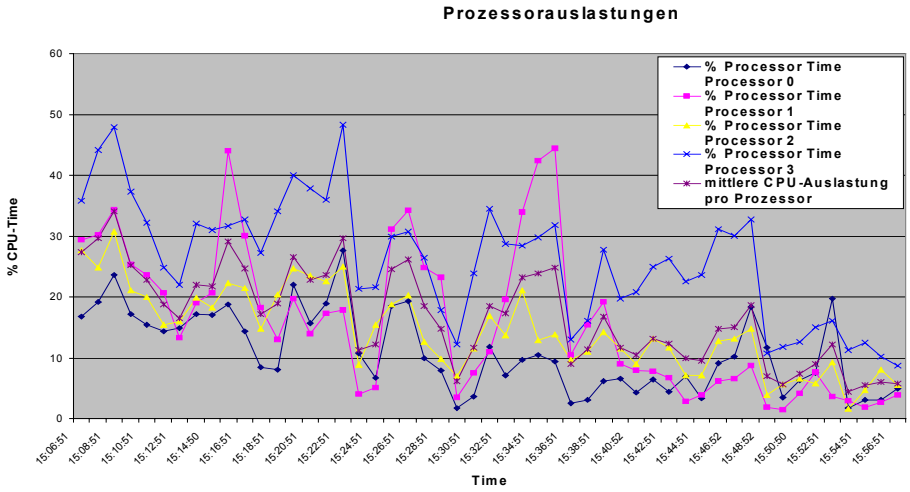


Abbildung 02-4: Prozessorauslastungen

Das zweite Diagramm zeigt die Auslastungskurven für die Anwendungen SAP und Oracle.

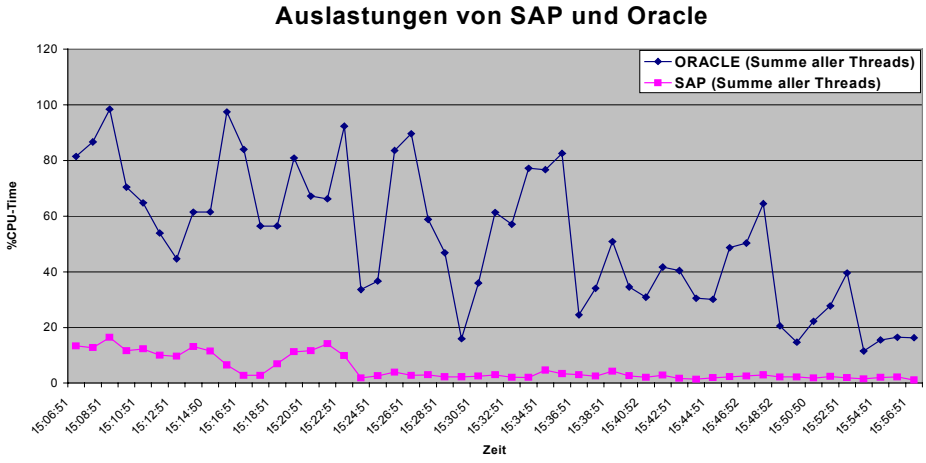


Abbildung 02-5: CPU-Belastung durch SAP und Oracle

Im dritten Diagramm werden die Paging-Aktivitäten in einem Zeitintervall von einer Stunde dargestellt.

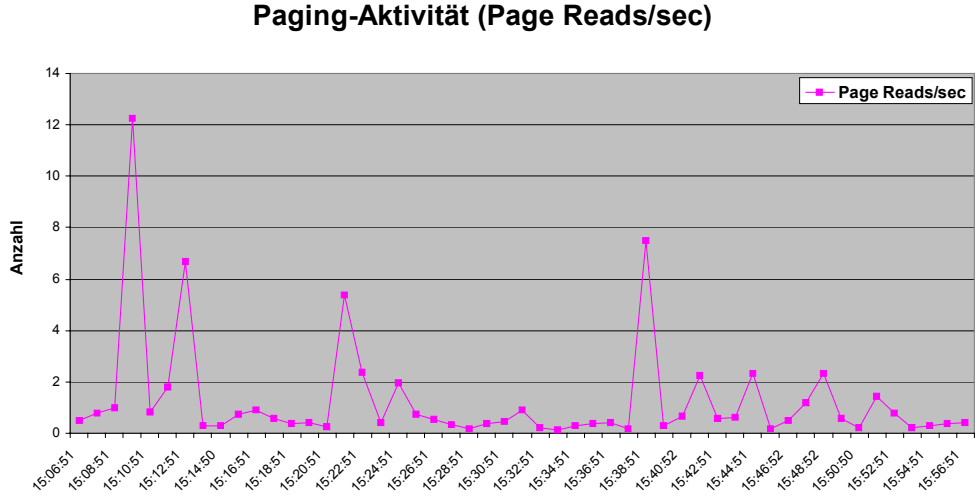


Abbildung 02-6: Paging-Aktivität

(c) Systemmonitore für Unix und Windows NT

Für die Betriebssysteme Unix, Linux und Windows NT bietet Siemens die PERMEV-Produktfamilie zur Überwachung der Systemressourcen an:

- @ PERMEX (SINIX/RELIANT UNIX)
- @ PERMENT (WINDOWS NT)
- @ PERMELIX (LINUX)
- @ PERMELIS (SOLARIS)

Die Bezeichnung „PERME“ steht hierbei als Abkürzung für „PERformance Monitoring and Evaluation Environment“ mit anschließender Abkürzung für das jeweilige Betriebssystem. Die genannten Produkte können zur automatischen und synchronisierten Aufzeichnung von Systemressourcenbelegungen und -auslastungen genutzt werden. Für die Messung werden die auf dem System vorhandenen „Standard“-Messwerkzeuge, wie zum Beispiel iostat, sar oder ps für Unix und der Systemmonitor bzw. die Performance Registry für Windows NT genutzt. Es erfolgt anschließend eine Auswertung der gewonnenen Daten. Die zum Beispiel mit Hilfe des Werkzeugs SAR möglichen Messergebnisse beschreiben das I/O-System oder, wie in Tabelle 02-7 dargestellt, die Prozessorauslastungen. Die Auslastung wird hierbei in Anwender- und System-Anteil unterschieden. Die Spalte „%wio“

gibt den Anteil der CPU-Zeit an, der für das Warten auf das I/O-System zur Verarbeitung der Aufträge benötigt wurde.

7.3.2000, 10:30-11:30	%usr	%sys	%wio	%idle
DB-Server	**21,98**	13,71	14,64	49,61
App-Server1	**7,34**	1,76	0,09	90,76
App-Server2	**20,95**	4,42	0,00	74,42
App-Server3	**32,75**	6,63	0,00	60,42
App-Server4	**30,82**	8,05	0,02	61,12

Tabelle 02-6: Prozessorauslastung im Stundenmittel (in Prozent)

Die Überwachung von Solaris-Maschinen kann mit Hilfe des frei erhältlichen Werkzeugs „SE Performance Toolkit“ von Richard Pettit und Adrian Cockcroft der Firma SUN vorgenommen werden (siehe [5], [6], [7]). Das Werkzeug basiert auf der SymbEL Language, eine an C orientierte Skriptsprache. Es können verschiedene Systemwerte und -parameter ausgelesen und teilweise auch gesetzt werden. Mit der Installation werden standardmäßig zwei ständig laufende Programme gestartet, die in vordefinierten Zeitintervallen das System beobachten, Daten sammeln und die Ergebnisse in Log-Dateien speichern. Der Vorteil dieses Werkzeugs liegt zum einen darin, dass es für jedermann frei erhältlich ist und zum anderen bereits viele Skripte enthält, die nahezu alle Betriebssystem-Daten abfragen. Diese Skripte können dann vom Anwender, den jeweiligen Bedürfnissen entsprechend, modifiziert werden.

(d) Datenbank-Überwachung

Die Datenbank-Überwachung kann mit Hilfe eines Datenbank-Monitors, wie zum Beispiel myAMC.DBMC (Application Management Center – Database Management Center) von Siemens, durchgeführt werden. Es werden zum Beispiel die folgenden Daten gesammelt:

- Cache Hit Ratio
- Physical Writes
- Physical Reads
- Number of requests (within the poll range)

- Number of transactions (within the poll range)
- Average utilization of the storage areas
- Number of locks

Die gemessenen Daten werden in eine Datenbank geschrieben. Falls der R/3 Live Monitor (myAMC.LNI) ebenfalls installiert ist, können beide Produkte die gleiche Datenbank verwenden. myAMC.DBMC unterstützt die Datenbanksysteme von Informix, Oracle und Microsoft. Eine detaillierte Beschreibung befindet sich in [8].

Zur Datenbank-Überwachung wird von den Herstellern bereits eine große Anzahl an Werkzeugen angeboten. Oracle enthält die „Oracle dynamic performance views“ (V$ views), die als Tabellen mit einer Vielzahl an statistischen Performance-Daten in der Datenbank verwaltet werden, wie zum Beispiel:

- V$SYSTEM_EVENT: View für die Wartesituationen im System
- V$SESSION-WAIT: View für die Wartesituationen innerhalb einer Session
- V$LOCK: View für die Locks im Datenbanksystem

Oracle bietet zwei Skripte UTLBSTAT und UTLESTAT zur Erstellung eines Reports über ein vordefiniertes Zeitintervall, der auf den V$-Views basiert. UTLBSTAT generiert einen „snapshot“ der V$-Views und UTLESTAT erzeugt einen zweiten „snapshot“ und berichtet über die Differenzen. Detaillierte Beschreibungen befinden sich in [9], [10], [11].

02.05 SAP Standard Applikations Benchmarks

(a) Motivation, Charakteristik, Nutzen

"Lügen - schlimme Lügen - Benchmarks". So sehr Benchmarks schon seit eh und je als unrealistisch und aussagelos verteufelt wurden, so unverzichtbar sind sie doch, wenn es darum geht, die richtige Konfiguration für ein konkretes IT Projekt zu finden. Unbestritten spiegelt ein Benchmark niemals die Wirklichkeit wider, doch gibt es genauso wenig nur *eine* Wirklichkeit. Ein Benchmark kann sozusagen als ein spezieller Anwendungsfall betrachtet werden und ist somit nicht unrealistischer als jede "wirkliche" Anwendung. Entscheidend für die Aussagefähigkeit eines Benchmarks ist, wie "nahe" er an der geplanten echten Anwendung liegt, inwieweit die im Benchmark erzeugte Last also der tatsächlich erwarteten Last entspricht. Ein Benchmark, der die spezielle Anwendung untersucht (ein "Applikationsbenchmark"), leistet zu dieser Überprüfung die besten Dienste. Alle großen Anbieter von ERP-Software haben deshalb Applikationsbenchmarks für ihre Anwendungen definiert. Davon zu unterscheiden sind die "Standardbenchmarks" wie TPC-C, SPEC

etc., bei denen eine abstrakte Last definiert wird. Letztere werden vor allem dazu verwendet, um die Hardware von unterschiedlichen Herstellern zu vergleichen. Der SAP Benchmark nimmt eine Sonderrolle ein, da er sich als Anwendungsbenchmark zu einem Standard etablieren konnte und damit die Vorteile beider Arten (Lastvorhersage und Hardwarevergleich) in sich vereint.

Durchgeführt werden die SAP Benchmarks in der Regel von den Hardwarepartnern der SAP. Sie verfolgen damit zwei Ziele: Einerseits werden mit Hilfe der Benchmarks unterschiedliche betriebswirtschaftliche Szenarien auf einer konkreten Hardware untersucht. Die so gewonnenen Daten werden verwendet, um die passende Hardwarekonfiguration zu jeder Kundenanforderung zu bestimmen. Zum anderen werden SAP Benchmarks im Wettstreit der Hardwarehersteller für Marketingzwecke verwendet. Das SAP Benchmark Council, ein Zusammenschluss von SAP und ihren Technologie Partnern, wacht darüber, dass dieser Wettstreit fair ausgetragen wird. Hierzu werden durchgeführte Benchmarks von der SAP "zertifiziert"– als Bescheinigung für die korrekte Durchführung eines Benchmarks. Alle zertifizierten Ergebnisse sind öffentlich zugänglich und somit von jedermann überprüfbar. Die Tatsache, dass es sich dabei auch um einen echten Anwendungsbenchmark handelt, macht die Ergebnisse in gewisser Weise wertvoller als die von rein synthetischen Standardbenchmarks. Beim SAP Benchmark werden alle Bereiche eines SAP Systems (CPU, Memory, I/O, Netzverkehr) zur gleichen Zeit getestet. Dies macht es praktisch unmöglich, mit Tricks besonders gute Ergebnisse zu erzielen. So ist es allein auf Grund des Umfangs nicht möglich, einen Compiler zu bauen, der speziell den SAP Code besonders gut übersetzt (wie z.B. bei SPEC üblich). Entsprechendes gilt für die Datenbank: Die vielen beteiligten Tabellen und die komplexen, internen Vorgänge würden es jedem Datenbankhersteller schwer machen, ihren Code speziell auf diese Last hin zu optimieren (wie z.B. bei TPC-D geschehen).

Zur Durchführung von SAP Standard Benchmarks werden Dialogbenutzer simuliert. Jeder simulierte Benutzer führt dabei eine Folge von vorgegebenen Transaktionen durch. Die Zeit zwischen zwei Eingaben, die "Denkzeit", ist mit 10 Sekunden fest vorgegeben. Ein simulierter Benutzer würde also einem sehr konzentriert und schnell arbeitendem echten Benutzer entsprechen. Die Zeit, die der Benutzer bei jedem Bildwechsel auf die Antwort vom System warten muss, die "Antwortzeit", wird im Benchmark protokolliert. Nach einer Regel muss diese Antwortzeit im Mittel unter zwei Sekunden liegen. Im Benchmark wird nun versucht, die Leistungsgrenzen eines Systems zu ermitteln. Dazu wird die Anzahl parallel simulierter Benutzer so lange erhöht, bis die mittlere Antwortzeit von unter zwei Sekunden erreicht wird. Ein wichtiges Ergebnis eines Benchmarks ist deshalb die Anzahl parallel arbeitender Benutzer und deren Antwortzeit.

Da der Begriff eines "Benutzers" jedoch in der Realität nur schwer zu greifen ist (ein "Benchmark Benutzer" ist genau definiert, unter einem "echten Benutzer" stellt sich jedoch jeder

jeder etwas anderes vor), wird neben der Benutzeranzahl auch der erzielte Durchsatz als wichtiges Ergebnis angegeben. Der Durchsatz wird dabei aus betriebswirtschaftlicher Sicht angegeben, wie z.B. "Anzahl der bearbeiteten Aufträge pro Stunde". Diese Sicht erlaubt es, das Ergebnis unabhängig von der technischen Durchführung zu interpretieren. Neben dieser Art von Benchmarks, in denen der Dialogbetrieb simuliert wird, gibt es noch eine zweite Art, bei der eine Hintergrundverarbeitung ("Batch") simuliert wird. Die angegebenen Ergebnisse sind dieselben, es wird jedoch keine Benutzeranzahl und Antwortzeit vermerkt.

Bei einem Benchmark werden folgende Ergebnisse angebenen.

Anzahl Benutzer	Parallel simulierte Benutzer mit einer Denkzeit von 10 Sekunden (nur Dialogbenchmark)
Antwortzeit	Die mittlere Antwortzeit (nur Dialogbenchmark)
Durchsatz	Durchsatz des Gesamtsystems in unterschiedlichen Einheiten: - Geschäftsvorfälle pro Stunde (abhängig von der Komponente) ➔ Die betriebswirtschaftliche Sicht - Dialogschritte pro Stunde ➔ Die technische Sicht - SAPS (nur bei SD) Die unterschiedlichen Durchsatzangaben können über konstante Faktoren ineinander umgerechnet werden.
Auslastung	CPU-Auslastung der Server
Konfiguration	2- oder 3-stufig, Anzahl Server etc.
Versionen	SAP, Datenbank, Betriebssystem
Plattenplatz	Insgesamt benötigter Plattenplatz

Tabelle 02-7: Angaben für Benchmarktest

Werden SAP Benchmarks zum Vergleich von unterschiedlichen Hardwareherstellern herangezogen, so dient der - unter der Restriktion einer Antwortzeit < 2 sec - erzielte Durchsatz als entscheidendes Leistungskriterium: "Hoher Durchsatz ist gleich gutes Ergebnis". Jedoch sind zwei beliebige Messungen nicht unbedingt vergleichbar. Um beurteilen zu können, welche Benchmarks vergleichbar sind, ist ein gewisses Grundverständnis erforderlich. Insbesondere muss man die Unterschiede bei der Konfiguration (2- oder 3-stufig), bei den verwendeten SAP Releases und bei den unterschiedlichen Benchmarkkomponenten (SD, ATO etc.) berücksichtigen.

Die Multi-Tier Internet Architektur des SAP Systems erlaubt in der Realität die unterschiedlichsten Konfigurationen: Die Applikation kann auf verschiedene Arten auf mehrere

Server verteilt werden. Bei einem Benchmark hat man prinzipiell dieselben Freiheiten in der Konfiguration, was zu einer unüberschaubaren Vielfältigkeit führen würde. In der Praxis haben sich jedoch zwei Konfigurationstypen bei den Benchmarks herausgebildet: Die "2-stufige" Konfiguration (two-tier oder auch "Zentralsystem" genannt), bei dem das gesamte SAP-System und die Datenbank auf einem einzigen Rechner laufen, und die "3-stufige" Konfiguration (three-tier), bei dem auf einem Rechner ausschließlich die Datenbank läuft und auf mehreren anderen Rechnern die SAP-Applikation.

Da bei einem SAP-System der Ressourcenverbrauch der SAP-Applikation um Faktoren größer ist als der der Datenbank, werden zweistufige Benchmarks von der Last der SAP-Applikation dominiert. Hier spielen Faktoren wie Netzwerk und I/O-System kaum eine Rolle, entscheidend ist die schnelle Abarbeitung der SAP-Last auf den Prozessoren.

Der größte Durchsatz lässt sich in einer dreistufigen Konfiguration erzielen. Hier wird normalerweise der Datenbankserver ein großes SMP-System sein und die Applikationsserver auf denen die SAP-Applikation läuft, können viele kleinere Rechner oder einige große Server sein. Als hochskalierbares System kann ein SAP-System durch Hinzufügen von weiteren Applikationsservern immer so konfiguriert werden, dass schließlich der Datenbankserver die Durchsatzgrenze bestimmt. Bei einem dreistufigen Benchmark ist es deshalb immer das Ziel, die Leistungsgrenze des DB-Servers zu ermitteln. Es handelt sich also immer auch um einen "Machbarkeitstest" - es wird ermittelt, welcher maximale Durchsatz überhaupt möglich ist. Bei einem großen Datenbankserver erreicht die Gesamtkonfiguration oft eine beeindruckende Größe. Es wurden schon Benchmarks mit mehr als 160 verschiedenen Applikationsservern durchgeführt. Aus einem solchen Benchmark lassen sich nicht nur Aussagen über die Leistungsfähigkeit der Datenbank bzw. des Datenbankservers gewinnen, sondern auch über die gesamte verwendete Konfiguration mit den Applikationsservern, dem Netzwerk und dem I/O System. Zwei- und dreistufige Benchmarks haben also sehr unterschiedliche Lastcharakteristiken und können nicht direkt miteinander verglichen werden.

Neben der Konfiguration muss bei einem Vergleich auch die SAP-Version beachtet werden. SAP als "lebendes" System wird stets weiterentwickelt und gewinnt dabei an Funktionalität, was aber auch teilweise bei neueren Versionen zu einem erhöhten Ressourcenverbrauch führt, abhängig vom betriebswirtschaftlichen Prozess. Sofern dieser Unterschied nicht genau bekannt ist, können Benchmarks mit unterschiedlichen Versionen nur sehr bedingt vergleichen werden. Generell gilt, dass Komponenten wie SD bei neuen Versionen meist ressourcenintensiver werden. Andere Komponenten, wie z.B. Retail oder BW können in neueren Versionen noch signifikante Performance-Optimierungen beinhalten.

Als drittes muss die "Benchmark-Komponente" beachtet werden. Der SAP Standard Applikations Benchmark ist eigentlich nicht nur *ein* Benchmark, sondern er besteht aus mehreren getrennt zu betrachtenden Komponenten. Unterschiedliche Komponenten decken dabei jeweils andere Anwendungsszenarien ab. Außerdem spiegeln unterschiedliche Komponenten die Entwicklung des SAP Benchmarks wider: Eine einmal getroffene Definition einer Komponente bleibt prinzipiell unverändert. Neue SAP-Funktionalitäten können deshalb nur in neuen Komponenten berücksichtigt werden. Besonders durch die Einführung von mySAP.com entsteht so die Notwendigkeit für neue Komponenten, die den Aspekten einer komplexen mySAP.com-Landschaft Rechnung tragen.

Die mySAP Supply Chain Management Szenarien werden von den Komponenten SD, ATO, MM, WM, PP und der geplanten Komponente APO abgedeckt. SD kommt eine besondere Bedeutung zu: Aufgrund der vielen Veröffentlichungen und der breiten Akzeptenz hat sich SD als echter Standard Benchmark durchsetzen können. SD-Benchmarks werden verwendet, um Systeme unterschiedlicher Hardwarehersteller zu vergleichen. Wie bei anderen Standardbenchmarks gibt es die SD-Top-Listen und den stets spannenden Kampf der Hersteller um einen der vorderen Plätze. Mit Hilfe des SD-Benchmarks wird außerdem die Einheit "SAPS" definiert .

SAP Application Benchmark Performance Standard

"100 SAPS sind definiert als 2000 vollständig verarbeitete Auftragspositionen pro Stunde im Standard SD Application Benchmark. Dieser Durchsatz wird erreicht bei 6000 Dialogschritten (Bildschirmwechsel) und 2000 Verbuchungen pro Stunde im SD Benchmark oder bei der Verarbeitung von 2400 SAP Transaktionen."

Die Definition von SAPS basiert auf dem betriebswirtschaftlichen Prozess und ist somit unabhängig von einer speziellen Konfiguration oder von der verwendeten Version. Die Einheit SAPS spielt eine wichtige Rolle bei der Beurteilung der Leistungsfähigkeit von Rechensystemen.

Abbildung 02-7: Die SAPS-Definition

Bei ATO wird durch ein komplexeres Customizing eine größere Anzahl Interaktionen zwischen unterschiedlichen SAP-Teilbereichen wie financials, materials management oder order fulfillment erreicht. Dabei werden Funktionalitäten wie z.B. Workflow oder der ATP-Server verwendet. Ob ATO den SD-Benchmark als Standard ablösen kann ist ungewiss – entscheidend hierfür wird die Akzeptanz bei den Hardwareherstellern und den Anwendern sein.

Nur wenige zertifizierte Benchmarks existieren für die mySAP Financials Komponente FI. Keinen zertifizierten Benchmark gibt es für das mySAP Product Lifecycle Management PS und für die mySAP Human Resources Komponenten HR und CATS. Auch ohne eine Zertifizierung anzustreben, werden solche Komponenten jedoch von den Hardware-Herstellern in den Labors durchgeführt, um spezielle Anwendungsszenarien zu untersuchen.
Als Branchenbenchmarks könnte man Retail, IS-U/CCS und BCA bezeichnen. Hier liegt der Fokus sehr stark auf der konkreten Applikation.

Die Komponente BW untersucht die mySAP Business Intelligence Szenarien. Diese Datawarehouse Anwendung wird vermutlich als wichtiger Bestandteil einer mySAP.com-Landschaft weiter an Bedeutung gewinnen.

ITS (auch als Online Store bezeichnet) war ein erster Ansatz in Richtung Internet Sales. Dieser Benchmark wird in Zukunft durch die geplanten mySAP Customer Relationship Managerment Komponenten ersetzt. Es sind drei CRM-Szenarien geplant: Internet Sales, Mobile Sales und Callcenter. Vermutlich werden die neueren mySAP-Komponenten in Zukunft weiter an Bedeutung gewinnen, um komplexe IT-Landschaften zu untersuchen.

Komponente	zertifizierte Benchmarks seit 1995	Dialog oder Batch	
SD	144	D	Sales and Distribution
ATO	16	D	Assemble To Order
SD Parallel	11	D	SD for parallel databases
Retail	5	B	Retail
BW	4	D+B	Business Information Warehouse
FI	4	D	Financials
IS-U/CCS	1	B	Customer Care and Service
BCA	1	D+B	Banks Customer Accounts
MM	1	D	Materials Management
PP	1	D	Production Planning
ITS	1	D	Internet Transaction Server
WM	0	D	Warehouse Management
PS	0	D	Project System
CATS	0	D	Cross Application Timesheets
HR	0	B	Human Resource
APO	geplant		Advanced Planning and Optimizing
CRM	geplant		Customer Relationship Management 3 Szenarien

Tabelle 02-8: Zertifizierte Benchmarks seit 1995

Die meisten Benchmarks wurden mit SD zertifiziert. Hier ist somit die Vergleichbarkeit am besten. Die Top 10-Listen für den SD-Benchmark, Release 4.x für zwei- und dreistufige Architekturen, sind dargestellt in Tabelle 02-9 und Tabelle 02-10.

Weitere Informationen zu allen Benchmarks finden sich unter [12], [13], [14].

1.	15080 SAPS	4.0B	01.03.00	Fujitsu Siemens PRIMEPOWER 2000, 64 CPU, Solaris, Oracle
2.	13720 SAPS	4.0B	31.03.00	Compaq AlphaServer GS320, 32 CPU, Tru64 UNIX, Oracle
3.	9150 SAPS	4.0B	31.03.00	Bull Escala WPC 2400, 24 CPU, AIX, Oracle
4.	8550 SAPS	4.0B	03.12.99	IBM ES/6000 S80, 24 CPU, AIX, DB/2
5.	5780 SAPS	4.5B	13.06.00	IBM AS/400-9406-840, 24 CPU, OS/400, DB/2
6.	2160 SAPS	4.0B	20.10.99	Fujitsu Siemens PRIMEPOWER 600, 8 CPU, Solaris, Oracle
7.	2120 SAPS	4.0B	30.11.99	Fujitsu Siemens PRIMERGY N800, 8 CPU, Windows NT, Oracle
8.	2120 SAPS	4.0B	10.12.99	Fujitsu Siemens PRIMERGY N800, 8 CPU, Windows 2000, Oracle
9.	2070 SAPS	4.0B	31.03.00	Compaq AlphaServer ES40, 4 CPU, Tru64 UNIX, Oracle
10.	1910 SAPS	4.0B	03.11.99	Unisys Aquanta ES5085, 8 CPU, Windows NT, MS SQL Server

Tabelle 02-9: Top 10 für SD, R/3-Release 4.x, zweistufige Architektur

1.	117680 SAPS	4.6B	28.11.00	Fujitsu Siemens PRIMEPOWER 2000, 64 CPU, Solaris, Oracle
2.	97730 SAPS	4.0B	31.03.00	SUN Enterprise 10000, 64 CPU, Solaris, Oracle
3.	83450 SAPS	4.0B	07.09.99	IBM RS/6000 S80, 24 CPU, AIX, Oracle
4.	52770 SAPS	4.6C	26.02.01	Unisys e-@ction ES7000, 16 CPU, Windows 2000, MS SQL Server
5.	38370 SAPS	4.0B	31.03.00	Compaq ProLiant 8000 6/700, 8 CPU, Windows 2000, MS SQL Server
6.	33530 SAPS	4.0B	17.02.00	Compaq ProLiant 8000 6/550, 8 CPU, Windows 2000, MS SQL Server
7.	33080 SAPS	4.0B	17.02.00	Unisys e-@action ES5085, 8 CPU, Windows 2000, MS SQL Server
8.	25300 SAPS	4.0B	01.09.98	IBM RS/6000 S70, 12 CPU, AIX, Oracle
9.	25070 SAPS	4.0B	20.10.98	Bull Escala RL 470, 12 CPU, AIX, Oracle
10.	24800 SAPS	4.0B	28.08.99	DataGeneral AV8500R, 8 CPU, Win NT, Oracle

Tabelle 02-10: Top 10 für SD, R/3 Release 4.x, dreistufig (angegeben nur der DB-Server)

(b) Beispiel eines großen SD-Benchmarks

Jede Realisierung eines SAP R/3-Systems im oberen Leistungsbereich wird folgendes Merkmal der Client/Server-Architektur in Rechnung stellen: die Zweiteilung in R/3-Applikation und Datenbank. Die von diesen Hauptkomponenten benötigten Ressourcen sind abhängig vom eingesetzten R/3-Modul, von der HW-Plattform und den Softwareversionen. Der Sales & Distribution (SD) Benchmark, mit dem Hersteller die Leistungsfähigkeit anzubietender Systeme typischerweise nachweisen, bildet den Datenfluss und Ressourcenverbrauch von R/3-Installationen modellhaft ab. Hier entfallen im aktuellen R/3-Release 4.6 etwa 91 % des CPU-Verbrauchs auf die R/3-Applikation und 9 % auf die Datenbank. Die beiden Komponenten sind in ihrer inneren Struktur sehr unterschiedlich. Mit einer heterogenen Infrastruktur kann diesen Unterschieden auch aus Kostengesichtspunkten optimal entsprochen werden.

Der Datenbankteil hat wegen des strengen Konsistenzbegriffs für den *einen* physikalischen Datenbestand einen naturgemäß monolithischen Charakter. Die in diesem Bereich bekann-

ten Techniken zur Clusterung sind sehr aufwändig. Gleichzeitig sind Leistungsfähigkeit und Skalierbarkeit moderner Server ausreichend, um den Bedarf dieser Komponente auch für höchste Transaktionsraten abzudecken. Der klassische Datenbankserver ist somit ein großes SMP-System, das allenfalls zum Zweck der Hochverfügbarkeit in einem Zweifachcluster abgesichert wird. Dieser Server wird mit einer größeren Zahl Netzwerkkarten und mit einem stattlichen Storage-Subsystem ausgestattet sein. Die IO-Raten sind sehr hoch, wobei das Netzwerk deutlich mehr IO-Last erzeugt als die Platte. Bei dem Benchmark von Fujitsu Siemens Computers hatte der Datenbankserver des R/3-Systems 130000 Netzwerk-IOs pro Sekunde zu bewältigen. Alles wird darauf ankommen, dass HW-Interconnect, Datenbanksoftware und Betriebssystem bis zu einer hohen Anzahl an Prozessoren skalieren (Scale-up).

Demgegenüber eignet sich die R/3-Applikation zum Scale-out. Die vom R/3-System zu bewältigende Gesamtlast an SAP-Transaktionen lässt sich auf eine konfigurierbare Anzahl von Warteschlangen aufteilen. Jede Warteschlange wird von einer konfigurierbaren Anzahl von Arbeitsprozessen bearbeitet. Abgesehen von der Datenbankschnittstelle, wo eine Konkurrenz um Ressourcen stattfindet, arbeiten diese Prozesse weitgehend autonom. Es gibt einen mäßigen R/3-internen Kommunikationsbedarf, weil die Verbuchung von den Dialogprozessen an spezielle Verbuchungsprozesse übergeben wird. Außerdem gibt es eine globale R/3-Sperrverwaltung. Ein Netzwerkstandard wie TCP/IP über Ethernet, über den auch die Datenbankschnittstelle bedient wird, ist für diesen Kommunikationsbedarf jedoch ausreichend. Es ergibt sich eine bequeme Parzellierbarkeit der R/3-Applikation in voneinander weitgehend unabhängige Instanzen. Für deren Realisierung bieten sich kostengünstige Commodity-Server an, etwa Intel-basierte Vierwegesysteme. Die geringen IO-Anforderungen solcher Instanzen, die durch Rechenintensität gekennzeichnet sind, lassen sich mit einigen wenigen PCI-Anschlüssen erfüllen. Vor allem ergibt sich eine bausteinhafte Flexibilität für veränderlichen Leistungsbedarf, wie er durch funktionales Wachstum bedingt sein kann. Ein anderes Beispiel für den Wunsch nach Flexibilität ist das Outsourcing-Geschäft, wo ein Rechenzentrum eine Vielzahl von Kunden bedient.

Mit dem Betriebssystem Linux lässt sich auf Rechenknoten dieser Art eine sehr gute automatisierte Administrierbarkeit erreichen. Sie beginnt mit netzwerkbasierten Installationsverfahren, wo Betriebssystem und Softwarepakete von einem zentralen Installationsserver geladen werden, und schließt eine skriptbasierte Steuerung und Überwachung von R/3 ein. Alles in allem fügt sich die beschriebene Organisation der R/3-Applikation gut in Infrastrukturstrategien ein, die auf Standards, Flexibilität und Automatisierbarkeit setzen.

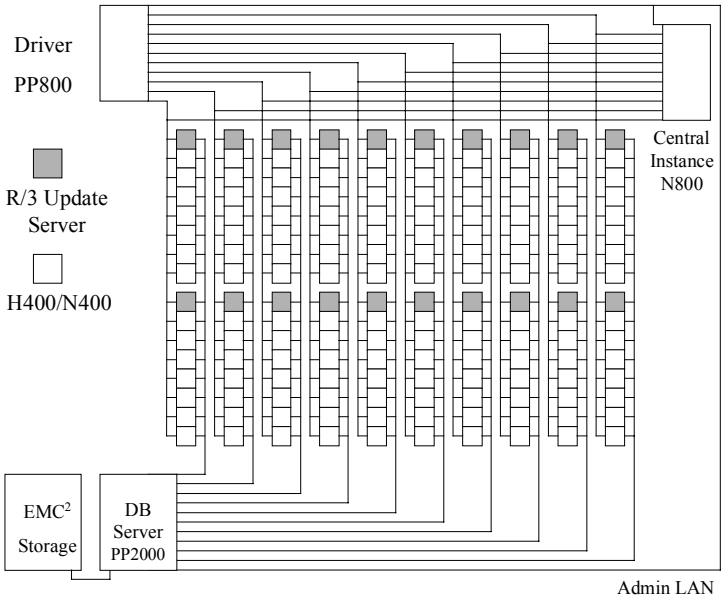


Abbildung 02-8: R/3-Installation für SD-Benchmark mit PRIMEPOWER

Die Abbildung 02-8 zeigt den Aufbau, mit dem Fujitsu Siemens Computers im Herbst 2000 die Realisierbarkeit eines SAP R/3-Systems für 23000 Benutzer des SD-Moduls nachgewiesen hat. Dieses Ergebnis ist von SAP zertifiziert und stellt die Weltbestleistung dar. Verwendung fand dabei bereits das R/3-Release 4.6, während die konkurrierenden Ergebnisse von Sun (19360 Benutzer), IBM (16640), Compaq (7500) und anderen noch das Release 4.0 einsetzten. Der Datenbankserver war eine PRIMEPOWER 2000 mit 64 CPUs vom Typ SPARC64 unter dem Betriebssystem Solaris und der Datenbank Oracle8i. Im Verlauf des Projekts hatte sich eine durchgehend gute Skalierung dieses Datenbankservers mit Faktor 1.9 bei CPU-Verdopplung erwiesen. Mit Servern bis 128 CPUs, die für das 2. Quartal 2001 angekündigt sind, lässt sich die Leistung von SAP R/3-Systemen nach dem hier beschriebenen Schema somit weiter ausbauen.

Dieses von Fujitsu Siemens Computers implementierte R/3-System unterstützte als SAP-zertifizierte Weltbestleistung 23000 SD-Benutzer. Die wesentlichen Komponenten waren eine PRIMEPOWER 2000 unter Solaris und Oracle8i als Datenbankserver, 160 Intel-basierte R/3-Applikationsserver vom Typ PRIMERGY mit jeweils 4 CPUs und eine R/3-Zentralinstanz vom Typ PRIMERGY mit 8 CPUs. Alle PRIMERGY-

Systeme liefen unter dem Betriebssystem Linux. Die Systeme waren durch 21 Ethernet-Netzwerke unter TCP/IP vernetzt.

02.06 Beispiele für IST-Analysen

Anhand der folgenden Beispiele wird gezeigt, wie durch systematisches Messen Performance-Engpässe mit den vorstehend beschriebenen Verfahren und Werkzeugen aufgedeckt werden können.

(a) Ressourcenmehrbedarfe durch Release-Wechsel

Der Release-Wechsel einer Software beinhaltet jeweils die Erweiterung des Funktionsumfangs und einer damit verbundenen Änderung des Ressourcenbedarfs. In den meisten Fällen steigen im R/3-Umfeld bei einem Release-Wechsel die Hardware-Anforderungen in erhöhtem Maße, so dass der Umstieg meist Hardware-Erweiterungen nach sich zieht. Die zu erwartenden Ressourcenmehrbedarfe können mit Hilfe von Benchmarks der Standard-Module verifiziert werden.

Bei einem Vergleich der Benchmark-Ergebnisse der unterschiedlichen R/3-Releases erhält man für die CPU-Mehrbelastungen eines Release-Wechsels von Version X.3[19] auf X.4 zum Beispiel die Tabelle 02-11. Hier werden modulspezifische Faktoren für die CPU-Mehrbelastungen dargestellt, mit denen die prozentualen Aufschläge berechnet werden. Es wird zum Beispiel für FI unter X.4 47 % mehr CPU-Leistung im Dialog-Betrieb beansprucht als unter dem R/3-Release X.3. Man erkennt, dass sich die Performance-Veränderungen in den unterschiedlichen Modulen sehr stark unterscheiden. Für die einzelnen Module könnten sogar für die einzelnen Transaktionen nach Angaben der SAP Leistungsunterschiede angegeben werden, da für einen Release-Wechsel nicht alle Transaktionen modifiziert worden sind.

[19] X.3 und X.4 stehen hierbei als Platzhalter für R/3-Releases.

R/3-Modul	DIA	UPD	DB
FI	1,47	1,00	1,14
MM	1,13	1,03	1,03
PP	1,50	1,00	1,67
SD	1,48	1,01	1,20
WM	1,46	0,91	1,26

Tabelle 02-11: CPU-Mehrbelastungen auf Grund eines Release-Wechsels

(b) Messung der vom Kunden eigenentwickelten ABAP/4-Anwendungen

Die Historie aller relevanten Leistungsdaten einer R/3-Installation werden in einer Performance-Datenbank abgelegt, die zum Beispiel durch SAPs Early Watch und Going Live Check genutzt [1] wird. Beide Reports geben Hinweise zur optimalen Parametrisierung des Systems, zur Performance-Verbesserung und zu möglicherweise zukünftig auftretenden Engpässen. Der Going Live Check wird nach Neu-Installation eines R/3-Systems einmalig ausgeführt, wobei der Early-Watch beliebig oft durchgeführt werden kann.

In den Reports werden für die einzelnen Server CPU- und Hauptspeicher-Auslastungen angezeigt. Es werden für die R/3-Workload, Datenbank-Last und Gesamtantwortzeit der Transaktionen die Verursacher des prozentual größten Anteils der Last dargestellt. Anhand der Transaktionen können die Programme bzw. R/3-Module identifiziert werden, die einer genaueren Betrachtung bedürfen. Für die einzelnen Module werden die prozentualen Anteile an der CPU- und DB-Last berechnet und grafisch präsentiert. Hierbei werden auch die eigenprogrammierten Anwendungen als eine Gruppe „Customer“ berücksichtigt.

Mit Hilfe der Filterfunktionalität des R/3 Live Monitors (Significant Dialogsteps) können die in den Reports dargestellten Lastverursacher für zukünftige Messungen getrennt von der Gesamtlast betrachtet werden. Es können für die gefilterten Dialogschritte bzw. Anwendungen TCQ-Profile zur Analyse der gemessenen Last erstellt werden. Zum Beispiel sollte im Dialogbetrieb der Datenbankzeit-Anteil der Dialogschritte an der Gesamtantwortzeit 40 % nicht übersteigen, da ansonsten ein Datenbankproblem vorliegen könnte. Weitere Richtwerte zur Interpretation der Antwortzeiten werden in [2] dargestellt.

(c) Datenbank-Tuning

In einem SAP R/3-System werden in der Praxis zentrale Datenbanken verwendet, die nicht über mehrere Server verteilt werden können. Die Datenbank wird somit häufig zum Per-

formance-Engpass des gesamten Systems. Eine wichtige Rolle spielt somit das Monitoring und Tuning der Datenbank.

Die Analyse einer Datenbank kann mit Hilfe der Datenbank-Werkzeuge und SAP-Reports erfolgen. Der SAP-Report „Going Live Check“ liefert zum Beispiel die folgenden Aussagen:

@ Überprüfung der Parametereinstellungen der Datenbank (z.B. Puffer-Einstellungen)

@ Überprüfung der Indexe (z.B. fehlende Indexe, die im R/3-Data Dictionary definiert sind, jedoch nicht in der Datenbank übernommen wurden)

@ Auflistung „teurer“ SQL-Anweisungen

- Es werden die SQL-Anweisungen und deren Anteil an der Gesamt-DB-Last beschrieben.
- Mit Hilfe der EXPLAIN-Funktion in Oracle werden Zugriffspfade der SQL-Anweisung dargestellt und welche Indexe für die Anfrage verwendet werden.
- Die DB-Last einer SQL-Anweisung wird anhand der gemessenen Datenbank-Puffer-Zugriffe in Relation zur Gesamt-Anzahl definiert, z.B. „The database load that is caused by a statement can be measured in the number of database buffers which the database system must access to execute the statement. This statement causes 16.62 % of all database buffer read accesses. It was executed 1362 times with an average of 118948.84 database buffers accessed per execution.“
- Es wird die Anwendung ausgegeben, die die SQL-Anweisung enthält.
- Der Report bietet Verbesserungsvorschläge zur Optimierung der SQL-Anweisung, zum Beispiel Definition eines weiteren Index, Erweiterung der WHERE-Klausel mit weiteren Attributen oder auch eine verbesserte Verwendung des Datenbank-Optimierers für die Wahl der Zugriffspfade

Weitere Möglichkeiten der Datenbank-Analyse bieten die Werkzeuge der Datenbankhersteller, wie zum Beispiel UTLBSTAT-UTLESTAT von Oracle, die über ein Zeitintervall einen Performance-Bericht der V$-Tabellen erstellen (Burleson 1998). Auch die Verwendung des R/3 Live Monitors kann zur Beobachtung und Analyse der Datenbank beitragen. Mit Hilfe der TCQ-Profile können anhand der Database-Service-Units die zugehörigen Datenbankverweilzeiten betrachtet werden. Bei einer niedrigen Anzahl an DBSU und einer hohen Datenbank-Verweilzeit sind Wartezeiten, wie zum Beispiel Locks, zu vermuten.

Mittels der gemessenen Einzelstatistiken können im R/3 Live Monitor auch die Datenbankaktivitäten (Read, Insert, Delete und Update) der Dialogschritte betrachtet werden.

02.07 Ausblick

Die beschriebenen Methoden und Vorgehensweisen werden im Rahmen von Kundenprojekten erprobt, validiert und erweitert. Die Weiterentwicklungen werden durch die beim Kunden vorhandenen Hardware- und Software-Ausstattungen, wie zum Beispiel R/3-Releases, RAID-Systeme oder auch neue Betriebssystem- oder DB-Versionen und die daraus resultierenden Anforderungen, vorangetrieben. Ein künftiges Ziel ist, die für SAP R/3 vorgestellten Vorgehensweisen auf mySAP.com zu übertragen. Das in diesem Beitrag beschriebene R/3-Performance-Engineering ist die Grundlage für die R/3-Kapazitätsplanung, die auf Basis der gewonnenen Daten Prognosen für zukünftige Szenarien, wie zum Beispiel Release-Wechsel oder Lastzuwächse, erstellt. Die Prognosen auf der Grundlage von Modellen aus der Warteschlangentheorie liefern Ergebnisse in Form von Auslastungen, Durchsätzen und Antwortzeiten.

02.08 Literatur

[1] Buck-Emden, Rüdiger; *„Die Technologie des SAP R/3-Systems“;* Addison Weley, 1999

[2] Schneider, Thomas; *„SAP R/3-Performance-Optimierung“;* Addison Wesley, 1999

[3] Dapper, Thomas; „Windows NT 4.0 im professionellen Einsatz“; Band 1, 1997

[4] Friedman, Marc; *“*Windows 2000 Performance Guide”, Mark Friedman, O'Reilly & Associates, Inc.; ISBN: 1565924665

[5] http://www.sun.com/sun-on-net/performance/se3/, Homepage für das SE Performance Toolkit

[6] “Server-Tuning mit dem SE Performance-Toolkit”; CT Nr. 6, 1999, Seite 275

[7] Cockcroft, Adrian; Pettit, Richard; *“Sun Performance and Tuning, Java and the Internet”;* Sun Microsystems Press, Prentice Hall, 1998

[8] Siemens ATD IT PS Mhm, *“myAMC.DBMC Handbuch“*

[9] Oracle; *„Oracle 8i On-Line Generic Documentation”;* Oracle Technology Network, http://www.technet.oracle.com/products/oracle8i

[10] Burleson, Don; *„Oracle8 Tuning”;* Sybex-Verlag, 1. Auflage, 1998

[11] Crain, Pat; Hanson, Craig; „Oracle Performance Analysis Using Wait Events“

[12] SAP; *“Standard Application Benchmarks - Published Results”;* Version 4.7, Januar 2001; http://www.sap.com/solutions/technology/pdf/50020428.pdf

[13] “The SAP Standard Application Benchmark Certified Results”; http://www.xware.net/sapbench

[14] Ideas International; *“Benchmarks, Top Performers Lists”*, http://www.ideasinternational.com/benchmark/bench.html

KAPITEL 03 MODELLIERUNG UND PROGNOSE FÜR SAP R/3-SYSTEME MIT DEM WLPSIZER

KAY WILHELM

03.01 Einleitung

Die betriebswirtschaftliche Standardsoftware mySAP.com wird von vielen Firmen zur Abwicklung und Verarbeitung ihrer Geschäftsprozesse eingesetzt. Ein Systemausfall oder eine schlechte System-Performance, wie z.B. erhöhte Antwortzeiten, können weitreichende Folgen für die Firmen nach sich ziehen. Produkte können nicht rechtzeitig fertiggestellt werden oder Mitarbeiter werden durch das System in ihrer Arbeitsgeschwindigkeit gebremst. Derartige System-Engpässe werden zum Teil durch einen Anstieg der Benutzerzahlen oder einen R/3-Release-Wechsel bzw. Lastzuwächse durch Installation weiterer Business Components (R/3-Module) verursacht. Das Ziel muss somit sein, derartige Szenarien frühzeitig zu erkennen und bereits im Voraus Systemengpässen entgegenzuwirken. So müssen zum Beispiel rechtzeitig notwendige System-Upgrades vorgenommen werden oder Server ausgebaut bzw. vervielfacht werden. Für die Erstellung von Performance-Prognosen wird in diesem Beitrag das Modellierungswerkzeug WLPSizer vorgestellt. Im Anschluss an die Werkzeugbeschreibung werden Vorgehensweisen und Beispiele für die Betrachtung der genannten Szenarien beschrieben.

03.02 Modellierungswerkzeug WLPSizer - Architektur

Der WLPSizer ist ein Werkzeug zur Modellierung von SAP R/3-Systemen. Die Abkürzung WLPSizer steht für „Workload Profile Sizer“. Ein zentraler Bestandteil des WLPSizers ist die Beschreibung der R/3-Last in Form von Workoad-Profilen, die automatisch anhand von Messdaten erstellt werden können. Der Begriff Sizer steht stellvertretend für die Möglichkeit Prognosemodelle zu erstellen, die Informationen bez. der in einem R/3-System zu erwartenden Performance und Hardware-Ausstattung liefern.

In diesem Abschnitt wird ein Überblick über die Architektur des WLPSizers gegeben. Es werden hierzu die in Abbildung 03-1 dargestellten WLPSizer-Komponenten, d.h. Bibliotheken und WLPMaker, sowie die Schnittstellen zu weiteren Werkzeugen, d.h. TOTO und MS Excel, beschrieben.

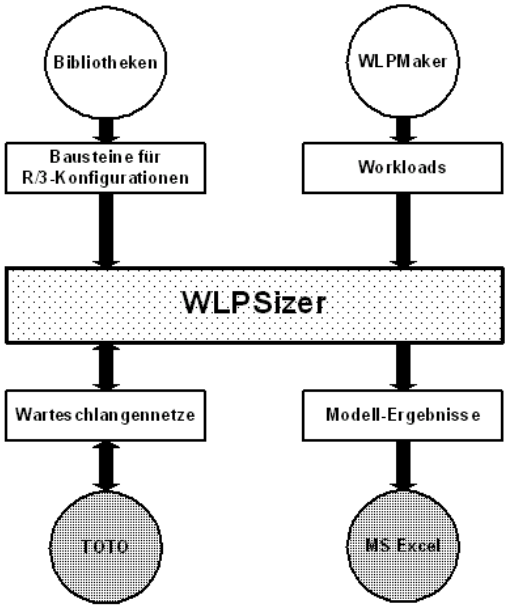


Abbildung 03-1: Architektur des WLPSizers

Zur Modellierung eines R/3-Systems mit dem WLPSizer benötigt man die Eingabe der R/3-Konfiguration (Beschreibung der Server, Netze, IO-Systeme) und eine Beschreibung der auf dem System befindlichen Last (Workload-Profile). Die Konfiguration wird mit Hilfe von Bibliotheken eingegeben. Wurde das R/3-System mit dem Werkzeug my-AMC.LNI[20] überwacht, können mit Hilfe des WLPMakers die Workload-Profile automatisch aus den Messdaten generiert werden. Die Bibliotheken und der WLPMaker sind, wie in der Abbildung 03-1 dargestellt, Bestandteile des WLPSizers.

Der WLPSizer kann nun anhand der eingegebenen Daten ein Warteschlangenmodell erstellen, das dem Warteschlangenlöser TOTO[21] übergeben wird. Die zurückgelieferten Modell-Ergebnisse können im WLPSizer dargestellt und nach MS Excel bzw. als ASCII-Datei exportiert werden. Im Folgenden wird auf die WLPSizer-Komponenten WLPMaker und die Bibliotheken sowie die Export-Möglichkeiten für die im WLPSizer erstellten Konfigurationen eingegangen.

[20] siehe [6]
[21] siehe [7]

(a) Bibliotheken

Eine für SAP R/3 typische 3-tier Konfiguration wird in Abbildung 03-2 dargestellt.

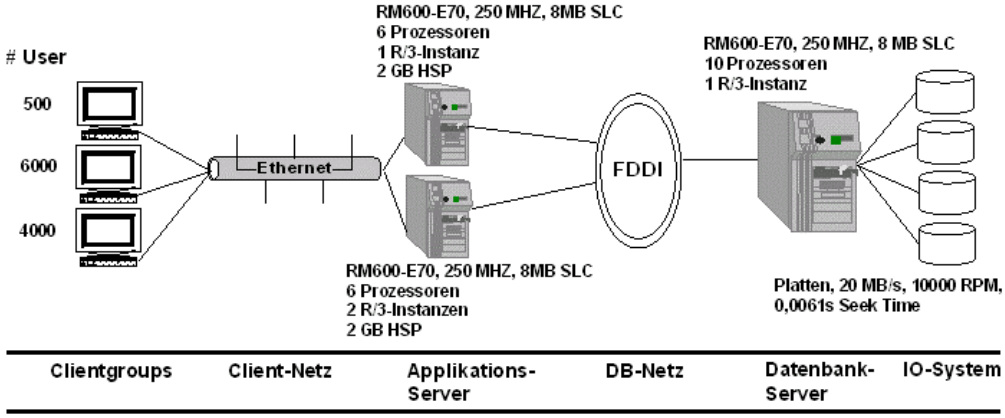


Abbildung 03-2: Beispiel für eine SAP R/3-Konfiguration

Die Basisstruktur eines R/3-Systems besteht aus den vier Grundelementen Clients, Netze, Application-Server und DB-Server. Der WLPSizer unterstützt die Modellierung von 3-tier und Zentralserversystemen, d.h. Applikation und Datenbank befinden sich auf einem Server. Die vier Grundelemente eines R/3-Systems können in beliebiger Anzahl, mit Ausnahme der DB, in den WLPSizer eingegeben und zu einem System verbunden werden. In der Praxis existiert in einem 3-tier R/3-System eine zentrale Datenbank, die sich auf einem Server befindet, so dass jedes WLPSizer-Modell nur einen DB-Server besitzen kann. Die Eingabe der einzelnen Systemkomponenten wird durch Bibliotheken unterstützt. Es existieren Bibliotheken für die Server, Netze und IO-Systeme.

CPU-Bibliothek

Die CPU-Bibliothek wird zur Spezifikation der Applikations- und Datenbankserver verwendet. Es können im WLPSizer verschiedene CPU-Bibliotheken angelegt werden, so dass die CPUs nach den verschiedenen R/3-Releases klassifiziert werden können. Dies ist notwendig, da die CPU-Leistung in SAPS[22] und somit in Abhängigkeit des R/3-Releases definiert wird. In Abbildung 03-4 ist der Dialog zur Bearbeitung der CPU-Bibliothek dargestellt.

[22] Siehe Kapitel 02 „Monitoring und Benchmarking für das R/3-Performance-Engineering".

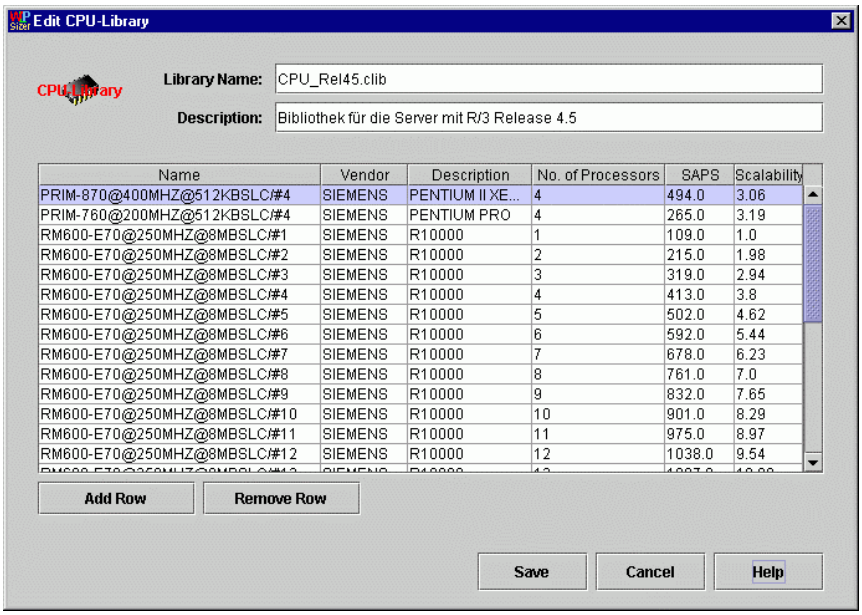


Abbildung 03-3: CPU-Bibliothek

Für jede CPU existieren die in der folgenden Tabelle 03-1 dargestellten Angaben:

Parameter	**Einheit**	**Beschreibung**
Name	---	Beschreibt alle für den CPU-Eintrag wichtigen Eigenschaften mit @ getrennt (Bezeichnung des Servers, Taktfrequenz, Second Level Cache, Anzahl der Prozessoren)
Vendor	---	Hersteller
Description	---	Beschreibung
No. of Processors	Anzahl	Anzahl der Prozessoren
SAPS	SAPS	Leistung der CPUs
Scalability	---	Skalierungsfaktor der Prozessoren, der das Ansteigen der Leistung beim Multiprozessorbetrieb beschreibt. Diese Steigerung ist in der Regel nicht linear, denn durch Overheads und seriell ablaufende Programme kommt es zu einer Verringerung der Leistung. Dieser Faktor wird durch den Quotienten der Mehrprozessor- und Einzelprozessor-Maschine berechnet. Skalierungsfaktor = SAPS (MonoProz) / SAPS (MultiProz)

Tabelle 03-1: Parameter der CPU-Bibliothek

Disk-Bibliothek

Im WLPSizer ist für den Datenbankserver neben den CPUs auch das Plattensystem mit anzugeben. Die Eingabe des IO-Systems erfolgt mit Hilfe der Disk-Bibliothek.

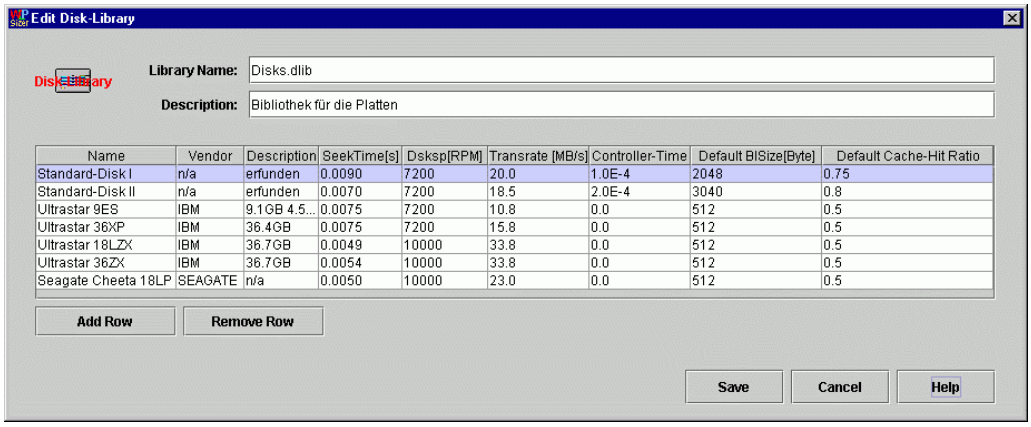


Abbildung 03-4: Disk-Bibliothek

Die Parameter der Disk-Bibliothek werden in Tabelle 03-2 beschrieben.

Parameter	Einheit	Beschreibung
Name	---	Bezeichnung des IO-Systems
Vendor	---	Hersteller
Description	---	Beschreibung des IO-Systems
Seek-Time	s	Mittlerer Zeitbedarf, der zur Positionierung der Schreib-/Lese-Köpfe über den gesuchten Zylinder benötigt wird
DskSp	RPM[23]	Rotationsgeschwindigkeit
Transrate	MB/s	Transferrate
Controller-Time	s	Zeitbedarf pro IO-Zugriff am Platten-Controller zur Überprüfung des Cache bzw. Lesen / Schreiben eines Blocks in oder aus dem Cache
Default BlSize	Byte	Blockgröße (Default-Wert, da der Wert nachträglich noch verändert werden kann)
Default Cache–Hit Ratio	---	Anteil der IO-Requests, die aus dem Cache bedient werden (Default-Wert, da der Wert nachträglich noch verändert werden kann)

Tabelle 03-2: Parameter der Disk-Bibliothek

[23] Rounds Per Minute

Ein Großteil der beschriebenen Parameter ist im Internet auf den Seiten der Hersteller frei erhältlich.

Protocol-Bibliothek

Zusätzlich zu den CPUs und IO-Systemen existiert im WLPSizer eine Bibliothek zur Eingabe von Netzen.

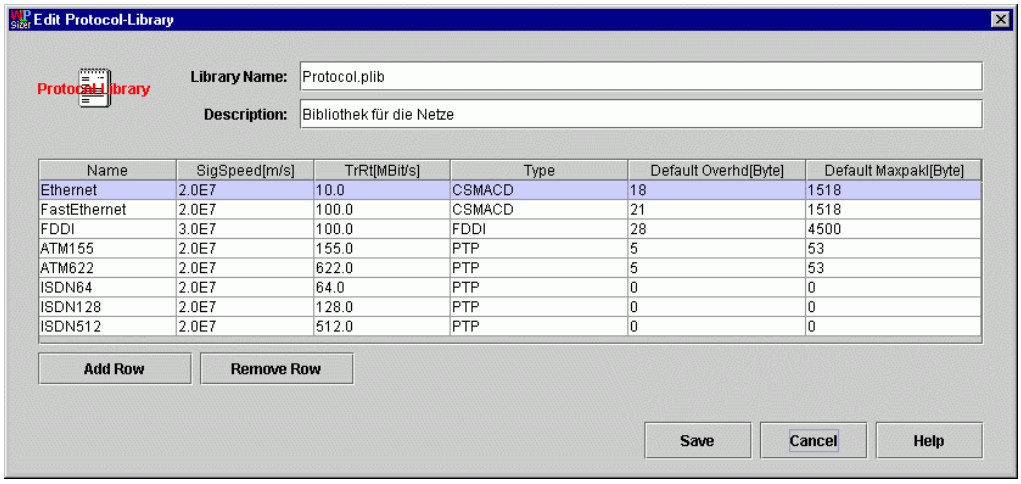


Abbildung 03-5: Protocol-Bibliothek

Die Protocol-Bibliothek enthält die in Tabelle 03-3 aufgeführten Parameter.

Parameter	Einheit	Beschreibung
Name	---	Bezeichnung des Netzes
SigSpeed	m/s	Signalgeschwindigkeit
TrRt	MBit/s	Bandbreite
Type	---	Netz-Protokoll
Default Ovhd	Byte	Vom Protokoll abhängiger Overhead (Default-Wert, da der Wert nachträglich noch verändert werden kann)
Default Maxpakl.	Byte	Vom Protokoll abhängige max. Paketlänge (Default-Wert, da der Wert nachträglich noch verändert werden kann)

Tabelle 03-3: Parameter der Protocol-Bibliothek

(b) WLPMaker

Der WLPMaker (Workload Profile Maker) ist ein Bestandteil des WLPSizers und wird zur Erstellung von Workload-Profilen verwendet. Die Workload-Profile werden auf Basis der vom myAMC.LNI gemessenen R/3-Last nach Eingabe der gewünschten R/3-Instanz und dem entsprechenden Zeitfenster automatisch generiert. Sie können anschließend direkt in das Modell als Lastbeschreibung eingegeben werden.

Konzept zur Beschreibung der R/3-Last mittels Workload-Profilen

Die R/3-Last wird durch die von den Dialogschritten verursachten Ressourcenverbräuche und Wartezeiten definiert. Nach [1] versteht man unter einem Dialogschritt die Verarbeitung von eingehenden Daten einschließlich einer Anfangsbildschirmmaske bis zum Abschicken eines Reaktionsdatenpaketes und der daraus resultierenden Antwortbildschirmmaske.

Das Werkzeug myAMC.LNI liefert neben vielen weiteren Ausgabe-Daten eine Darstellung der gemessenen Dialogschritte in Form von T60-Profilen. Es wird hierzu auf die Statistical Records des R/3-Systems zugegriffen, die alle verarbeiteten Dialogschritte mit ihren Ressourcenverbräuchen enthalten. In den T60-Profilen werden die Dialogschritte nach Tasktyp, Komplexitätsklasse und Dienstgüteklasse klassifiziert (siehe Kapitel 02 „Benchmarking und Monitoring für SAP R/3“). Das T60-Profil beschreibt typischerweise die R/3-Last einer R/3-Instanz.

In der nun folgenden Beschreibung der im WLPSizer verwendeten Workload-Profile müssen zwei Arten von Profilen unterschieden werden: Simple- und Extended-Workload-Profile. Wie in Abbildung 03- 6 dargestellt, ist ein Simple-Workload-Profil durch die Attribute CPUTI, DBSU, DBKB und ClBy definiert. Das Extended-Profil besitzt hingegen zusätzlich die Spalten RespTime, DBTime, LockTime und QueueTime. Beide Profile können vom WLPSizer eingelesen werden. Zur Modellierung werden jedoch nur die Spalten der Simple-Workload-Profile verwendet. Die Extended-Workload-Profile werden hingegen zur Kalibrierung des Modells herangezogen. Im weiteren Verlauf dieses Artikels wird unter der Bezeichnung Workload-Profil das Extended-Workload-Profil verstanden.

T	C	ratio	CPUTI	DBSU	DBKB	ClBy	RespTime	DBTime	LockTime	QueueTime
D	1	0,5228	19,50	72,84	0,57	325	87,77	20,36	0,14	0
D	2	0,1193	69,46	425,48	11,98	415	421,08	276,03	1,33	0
D	3	0,0322	932,84	1888,59	104,81	603	3959,03	2650,82	6,26	0
D	4	0,0045	6406,56	19454,60	2355,77	715	37959,57	28602,39	12,33	0
D	5	0,0007	73417,97	390184,00	123115,67	1015	379533,32	319135,92	0,11	2
U	1	0,0529	6,54	131,98	0,78	112	20,50	7,31	0,00	0
U	2	0,2076	28,51	595,96	31,14	336	88,35	40,38	0,28	10
U	3	0,0274	99,01	1568,43	191,28	645	676,18	543,27	0,33	6
U	4	0,0018	555,47	6360,15	456,03	812	3455,48	2912,10	0,24	0
U	5	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
B	1	0,0044	12,76	170,98	0,13	225	29,29	10,86	0,31	3
B	2	0,0015	43,20	414,47	20,71	336	607,08	214,62	0,86	2
B	3	0,0011	183,59	1411,25	52,95	548	2286,45	678,01	6,66	1
B	4	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
B	5	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
O	1	0,0143	12,21	59,86	0,00	96	366,25	1,58	0,00	313
O	2	0,0066	29,98	478,31	24,10	223	1606,70	29,55	0,00	8
O	3	0,0028	59,48	1467,97	30,83	469	2072,20	67,20	0,00	864
O	4	0,0001	93,75	5468,00	21,89	712	239,56	178,57	0,00	0
O	5	0,0000	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
*	*	**1,0000**	**139,59**	**707,05**	**114,79**	**344**	**734,87**	**516,10**	**0,56**	**9**

Abbildung 03-6: Workload-Profil

Die Abkürzungen im Workload-Profil werden in der folgenden Tabelle beschrieben.

Parameter	Einheit	Beschreibung
T	---	Tasktyp des Dialogschritts (Dialog, Update, Batch, Other)
C	---	Komplexitätsklasse (1 ... 5)
ratio	---	Anteil an der gesamten Workload
CPUTI	ms	mittlere normierte CPU-Zeit eines Dialogschritts im Workprozess[24]
DBSU	Anzahl	mittlere Anzahl der DBSU
DBKB	KB	mittlere Anzahl an Daten in KB, die zwischen DB- und Appl.-Server versendet wurden
OTHKB	KB	Summe der transferierten Bytes für ABAP/4-Quelltext und Programm laden sowie Dynpro/Screen-Quelltext und Dynpro/Screen laden
ClBy	Byte	mittlere Anzahl an Bytes, die zwischen Client und Application-Server transferiert wurden
RespTime	ms	mittlere Antwortzeit des Dialogschritts[25]
DBTime	ms	Verweilzeit des Auftrags in der Datenbank (gemessen an der Datenbankschnittstelle des Appl.-Server, somit inkl. Netzzeit zum DB-Server)
LockTime	ms	mittlere Sperrzeit im R/3-System (keine DB-Sperren)
QueueTime	ms	Wartezeit am Dispatcher
*	---	gewichteter Mittelwert der jeweiligen Spalte über alle Tasktypen und Komplexitätsklassen mit Ausnahme der Spalte „ratio“, die summiert wird

Tabelle 03-4: Beschreibung des Workload-Profils (Extended)

Es können mit Hilfe des WLPMakers beliebige Zeitintervalle als Workload-Profil dargestellt werden. Für die Modellierung werden Zeitintervalle von einer Stunde verwendet. Die Einträge des Workload-Profils entsprechen den mittleren Verbräuchen eines Dialogschritts im entsprechenden Zeitintervall. Wie bereits beschrieben, werden die Dialogschritte nach Tasktyp und Komplexitätsklasse gruppiert. Die Spalte ratio gibt den Anteil der jeweiligen Gruppe von Dialogschritten gemessen am Gesamtdurchsatz des Workload-Profils an. Für die CPU-Zeit werden die gemessenen Zeitverbräuche bez. eines Referenzrechners normiert, das heißt es wird berechnet, wie viel CPU-Zeit ein Rechner mit der Referenzleistung für die im Workload-Profil angegebene Last benötigt. Die Referenzleistung ist auf 250 SAPS festgelegt. Sie entspricht der Leistung eines Monoprozessor-Servers, wie zum Beispiel PRIMEPOWER M1000 (249 SAPS) mit SAP Release 4.6[26]. Die Normierung basiert

[24] Die CPU-Zeit beinhaltet ausschließlich die CPU-Zeit am Application-Server.

[25] Bis Rel. 4.6b wurde die Antwortzeit am Application-Server ohne Client-Zeit gemessen. Seit Rel. 4.6b wird auch die Client-Zeit gemessen.

[26] Benchmark-Ergebnisse der Firma Siemens.

auf den in SAPS angegebenen Leistungsdaten der Server. Für jede Zeile der Workload (20 Zeilen) wird die folgende Berechnung durchgeführt:

Für alle $j \in \{1,..,20\}$ gilt: $CPUTI_j^{WL} = CPUTI_j^{LNI} / \alpha$

mit folgenden Parametern:

Parameter	Beschreibung
$CPUTI_j^{WL}$	normierter CPU-Zeitverbrauch der Zeile j für das Workload-Profil
$CPUTI_j^{LNI}$	gemessener CPU-Zeitverbrauch für Zeile j
α	RP / (L/Skal)
RP	Referenzleistung (= 250 SAPS)
Svr	Server auf dem die Workload gemessen wurde
L	SAPS-Leistung von Svr
Skal	Skalierungsfaktor für die Anzahl der CPU von Svr

Tabelle 03-5: Parameter zur Normierung der Workload-Profile

Die Normierung der Workload-Profile liefert die Möglichkeit, sie in verschiedenen Konfigurationen wiederzuverwenden und miteinander vergleichen zu können.

Die Zeile „*“ liefert den gewichteten Mittelwert der jeweiligen Spalte mit Ausnahme der ratio, die aufsummiert wird. Berechnet wird der Mittelwert als Summe der Produkte von ratio und des jeweiligen Spalteneintrags:

$* = \Sigma_{i=1..20}$ (ratio * $Spalte_{ij}$)

mit i = aktuelle Zeile und j = aktuelle Spalte des Workload-Profils.

Anwendung des WLPMakers

Der in den WLPSizer integrierte WLPMaker dient zur Erstellung der bereits beschriebenen Workload-Profile. Es können hierzu die folgenden drei Funktionen ausgewählt werden:

- **Single Workload-Profile:** Es wird ein Workload-Profil für ein eingegebenes Zeitfenster erstellt.
- **Sequence Workload-Profile:** Es wird eine Sequenz von Workload-Profilen innerhalb eines eingegebenen Zeitfensters erstellt. Für die einzelnen Workload-Profile kann eine konstante Länge des zu beschreibenden Zeitintervalls angegeben werden. So kann zum Beispiel von 08:00 - 18:00 Uhr für jede Stunde ein Workload-Profil erstellt werden, das 60 Minuten der gemessenen Last beschreibt.

@ **TCQ-Profile:** Die Abkürzung TCQ steht für Tasktyp, Complexity und Service Quality. Das TCQ-Profil ist im Gegensatz zu den Workload-Profilen kein Eingabeparameter für das Modell, sondern ein Profil zur Darstellung der R/3-Last. Die Dialogschritte werden hierzu nach Tasktyp, Komplexitätsklasse und Dienstgüte gruppiert. Die Dienstgüte wird anhand der Antwortzeit des Dialogschritts definiert. Eine detaillierter Beschreibung der TCQ-Profile befindet sich in [2].

Exemplarisch wird der Dialog zur Erstellung eines Single Workload-Profils beschrieben (siehe Abbildung 03-7). Die weiteren Dialoge sind bis auf wenige Unterschiede identisch. Unter Database wird die ausgewählte myAMC.LNI-Datenbank angezeigt. Das Auswahlfeld Instance enthält alle gemessenen Instanzen des R/3-Systems. Hier kann man die Instanz auswählen, von der ein Workload-Profil erstellt werden soll. Werden für ein Zeitintervall Workload-Profile aller Instanzen benötigt, so kann die Funktion „Generate Profiles from all Instances“ aktiviert werden. Der Messzeitraum für die ausgewählte Instanz wird dann unter „Measured Time for this Instance“ dargestellt. Es kann daneben unter „Create Workload-Profile“ das von dem Workload-Profil zu beschreibende Zeitintervall eingegeben werden.

Für jedes Workload-Profil muss zur Normierung des Workload-Profils die CPU des Servers, auf dem die R/3-Last gemessen wurde, angegeben werden. Die Auswahl der CPU geschieht über die Schaltfläche „Select CPU...“. Anschließend kann die gewünschte CPU-Bibliothek geöffnet und die entsprechende CPU ausgewählt werden. Über den Reiter „Single WLP“ kann eingestellt werden, ob Simple- oder Extended-Workload-Profile erstellt werden sollen. Die Funktion „TT-separated“ bietet die Möglichkeit, nach Tasktypen getrennte Workload-Profile zu erstellen.

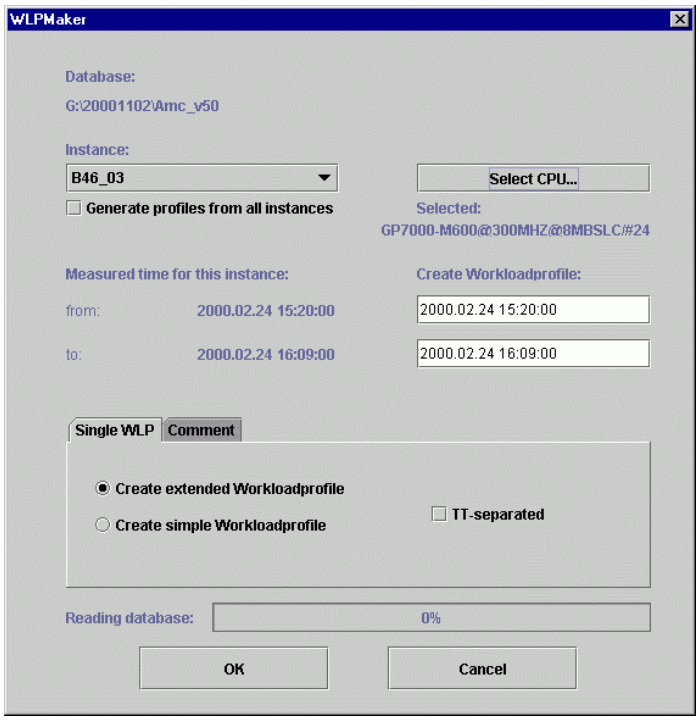


Abbildung 03-7: Erstellung von Workload-Profilen

(c) Export von Modellen und Ergebnissen

Zur Weiterbearbeitung der Modelle können diese in das Modellierungswerkzeug VITO übertragen werden. Weiterhin können in Form von Reports Modelle und Ergebnisse nach MS Excel exportiert werden.

VITO-Schnittstelle

VITO ist ebenso wie der WLPSIZER ein Modellierungswerkzeug basierend auf der Warteschlangentheorie, mit dem auch Nicht-SAP-Systeme betrachtet werden können (siehe [3]). Die im WLPSIZER erstellten Modelle können in das Programm VITO exportiert werden. Die Modelle werden in VITO grafisch dargestellt und können in VITO ohne weitere Nachbearbeitung direkt gelöst werden. In Abbildung 03-8 ist ein nach VITO exportiertes Modell dargestellt.

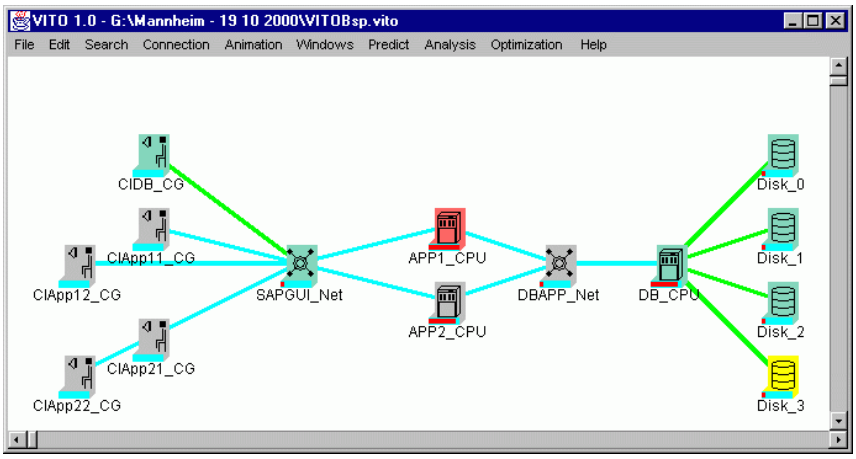


Abbildung 03-8: WLPSizer-Modell exportiert nach VITO

Reporting

Die Ergebnisse der Modelle werden in Form von Berichten zusammengefasst und dargestellt. Die Berichte können als Text-Datei gespeichert oder nach MS Exel überführt werden. Die Berichte können anschließend als MS Exel-Arbeitsmappe gespeichert werden (siehe Abbildung 03-9).

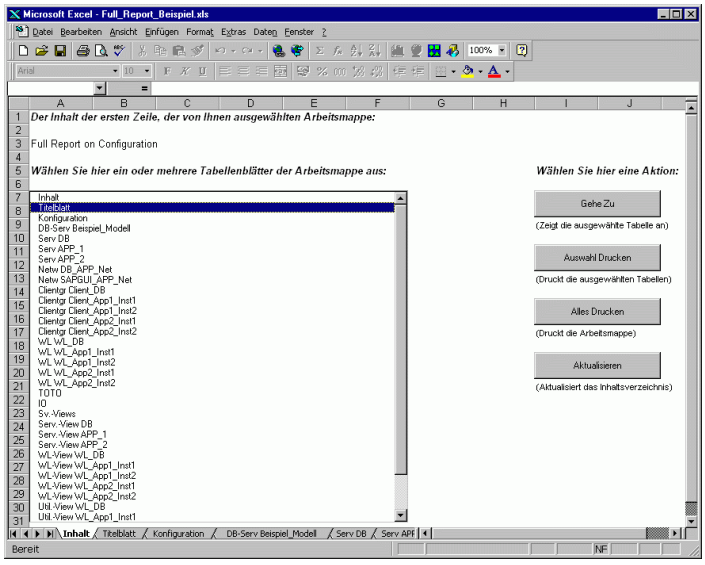

Microsoft Excel - Full_Report_Beispiel.xls
Der Inhalt der ersten Zeile, der von Ihnen ausgewählten Arbeitsmappe:
Full Report on Configuration
Wählen Sie hier ein oder mehrere Tabellenblätter der Arbeitsmappe aus:
Wählen Sie hier eine Aktion:
Gehe Zu
(Zeigt die ausgewählte Tabelle an)
Auswahl Drucken
(Druckt die ausgewählten Tabellen)
Alles Drucken
(Druckt die Arbeitsmappe)
Aktualisieren
(Aktualisiert das Inhaltsverzeichnis)

Abbildung 03-9: Report nach MS Excel exportiert

03.03 WLPSizer im Überblick

In diesem Abschnitt werden die Grundeigenschaften des WLPSizers beschrieben, die die Eingabe einer Konfiguration, das Lösen des Modells und die Darstellung der Ergebnisse ermöglichen.

(a) Konfiguration

Die im WLPSizer eingegebenen Modelle werden als Konfigurationen bezeichnet. Eine Konfiguration besteht aus den folgenden Komponenten:

- Datenbankserver und IO-System (DB-Server, IO-System)
- Applikationsserver (Application-Server)
- Netze (Nets)
- Benutzergruppen (Clientgroups)
- Lastbeschreibungen (Workloads)
- Konfigurationsparameter (Settings)[27]

Ein Application-Server ist dabei durch Prozessoren und ein DB-Server durch Prozessoren sowie IO-System charakterisiert. Bis auf den DB-Server können alle Komponenten beliebig oft in eine Konfiguration eingefügt werden. Es können somit nur Konfigurationen mit einer zentralen relationalen Datenbank betrachtet werden, die aber auch die in der Praxis gebräuchlichen R/3-Konfigurationen widerspiegeln (siehe [1], [4]).

Die Erstellung einer neuen Konfiguration im WLPSizer erfordert die Eingabe eines DB-Servers, eines Netzwerks und einer Clientgroup sowie den Namen der Konfiguration. Die dabei entstandene Konfiguration hat die in Abbildung 03-10 dargestellte Form und kann als Grundkonfiguration bezeichnet werden.

[27] Die Settings stellen keine typsiche Konfigurations-Komponente dar, werden jedoch mit jeder Konfiguration gespeichert.

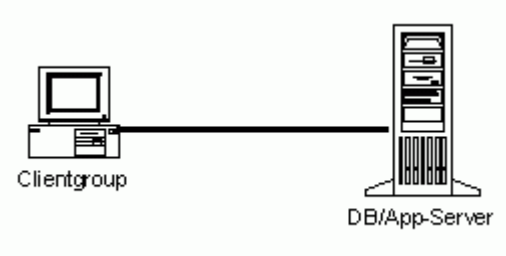


Abbildung 03-10: Grundkonfiguration

Wie in der Abbildung bereits dargestellt, wird der Server als DB- und Application-Server interpretiert, also als Zentralserver. Werden Application-Server der Konfiguration hinzugefügt, entspricht diese einer 3-tier R/3-Konfiguration. Die erstellte Konfiguration wird in der Config-View dargestellt (siehe Abbildung 03-11).

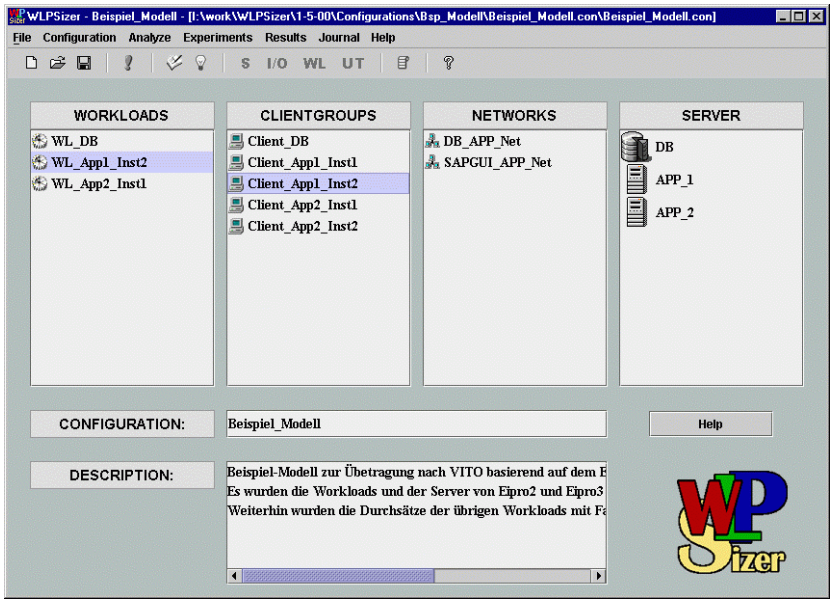


Abbildung 03-11: Config-View

Es werden in den folgenden Abschnitten die einzelnen Komponenten einer Konfiguration und deren Parameter beschrieben.

(b) Lastbeschreibungen – Workloads

Eine Workload repräsentiert die von einer R/3-Instanz oder einer bestimmten Applikation erzeugte Last. Sie wird durch die folgenden Parameter charakterisiert:

- *Bezeichnung:* Name der Workload
- *Profil:* Normiertes Workload-Profil
- *Durchsatz:* Anzahl der Dialogschritte pro Stunde

Zur Eingabe von Workloads erscheint der in Abbildung 03-12 dargestellte Dialog. Es sind der Name der Workload und die Anzahl der Dialogschritte pro Stunde (Durchsatz) einzugeben. Über einen Reiter kann der Workload-Typ ausgewählt werden. Es existieren im WLPSizer drei verschiedene Workload-Typen (Standard, Background, Custom), wobei in diesem Abschnitt nur die Standard Workloads beschrieben werden. Die weiteren Workload-Typen werden im Abschnitt „Erweiterte Funktionen und Vorgehensweisen zur Modellierung“ beschrieben.

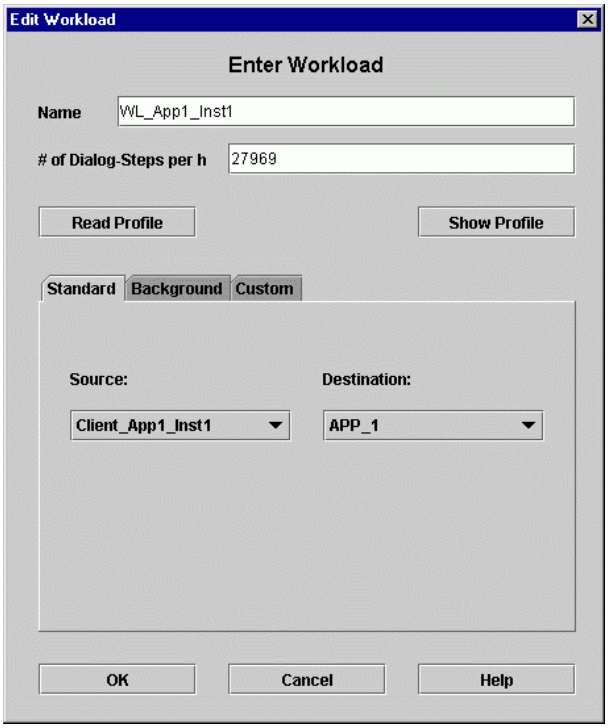


Abbildung 03-12: Workload-Dialog

Eine Workload vom Typ Standard besitzt eine fest vorgegebene Quelle und ein Ziel. Als Quelle kann eine Clientgroup (Lastverursacher) und als Ziel ein Server der Konfiguration ausgewählt werden. Das Ziel ist hierbei der Server, der die Instanz bzw. Applikation betreibt. Der WLPSizer bestimmt automatisch, welche Stationen der Konfiguration von der Workload besucht werden, so dass das Ziel erreicht wird.

Über die Schaltfläche Read-Profile kann ein Workload-Profil geladen werden, das der Workload zugrunde liegen soll. Name und Durchsatz der Workload werden automatisch dem Workload-Profil entnommen. Der eingetragene Durchsatz entspricht dem real gemessenen Durchsatz normiert auf eine Stunde. Die Schaltfläche Show Profile stellt das mit der Workload verbundene Workload-Profil dar (siehe Abbildung 03-13).

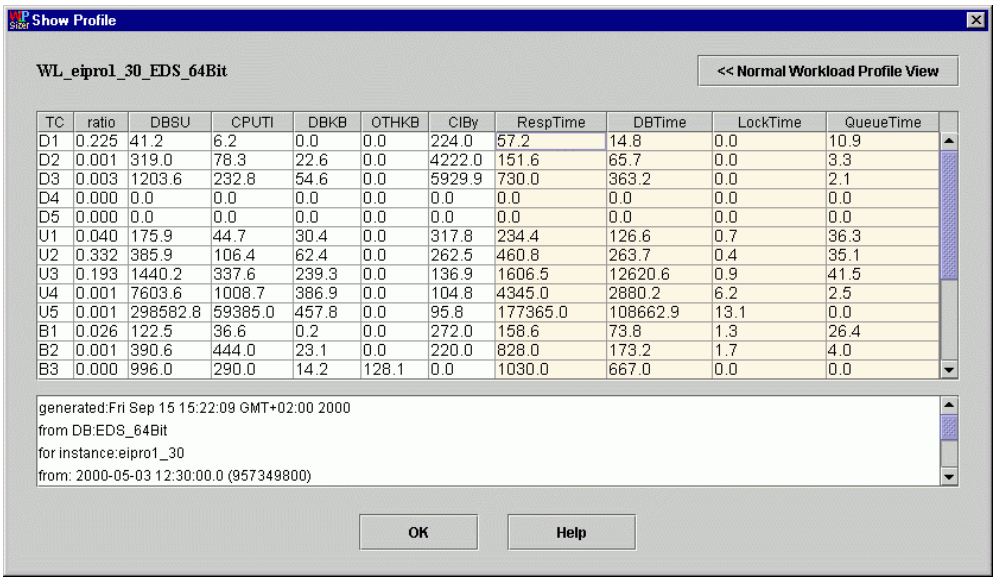


Abbildung 03-13: Workload-Profil-Dialog

Wurde für die Workload ein Extended Workload-Profil eingebunden, so kann über die Schaltfläche Extended Workload Profile die gesamte Workload angezeigt werden, wobei die im Vergleich zur Simple Workload zusätzlichen Spalten farbig markiert werden.

(c) Clientgroups

Eine Clientgroup repräsentiert eine Gruppe von Benutzern (bzw. Terminals) in einem gemeinsamen Netzsegment, die eine identische Last erzeugen. Eine Clientgroup wird parametrisiert durch:

@ *Active User:* Anzahl aktiver Benutzer

@ *Netz:* Netz, das die Clientgroup mit den Servern verbindet

Im WLPSizer erscheint zur Eingabe einer Clientgroup der in Abbildung 03-14 dargestellte Dialog. Es werden der Name und die Anzahl der aktiven Benutzer abgefragt. Aus einer Liste der im WLPSizer bereits definierten Netze kann das Netz ausgewählt werden, das die Clientgroup mit den Servern verbindet. Abschließend kann ein Kommentar in das hierzu vorgesehene Feld eingegeben werden.

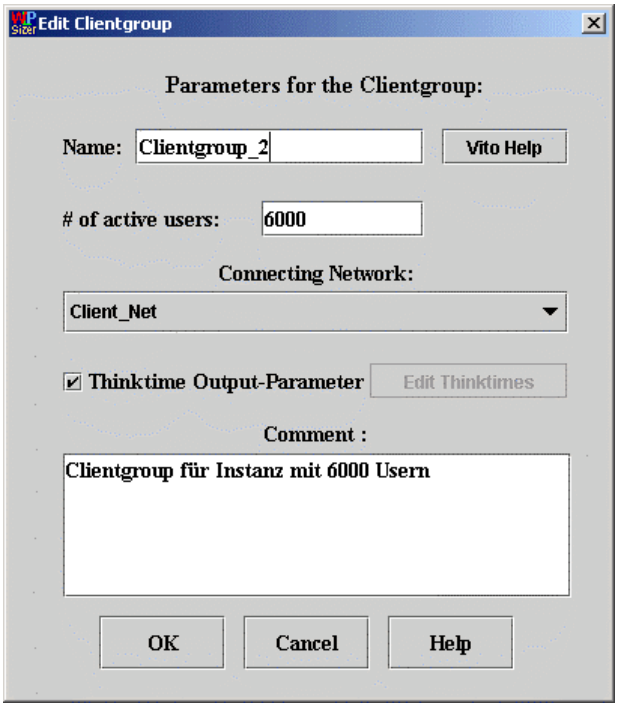


Abbildung 03-14: Clientgroup-Dialog

Im Modell besitzt jeder Benutzer eine Denkzeit, also die Zeit, die zwischen dem Abschluss und dem Neubeginn eines Dialogschritts vergeht. Wird der Kontrollkasten Thinktime Output –Parameter aktiviert, werden die Denkzeiten automatisch berechnet, so dass die Durchsätze der Workloads dieser Clientgroup erreicht werden (Durchsatzsteuerung). Ist der Kontrollkasten nicht aktiviert, so können eigene Denkzeiten pro Komplexitätsklasse eingegeben werden (siehe Abbildung 03-15). Die Denkzeiten gelten für alle Tasktypen und Workloads der Clientgroup. Die Durchsätze der Workload werden bei dieser Einstellung nicht erzwungen, so dass der jeweilige Durchsatz von der eingegebenen Anzahl der User abhängt.

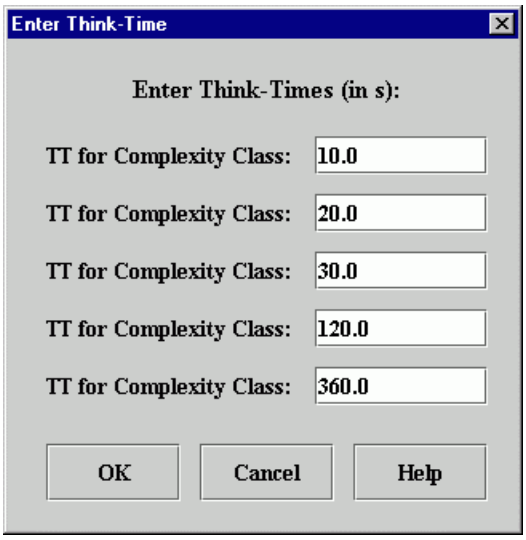


Abbildung 03-15: Denkzeiten-Dialog

(d) Application-Server

Die Application-Server werden im WLPSizer allein durch ihre CPUs charakterisiert. Die eventuell lokal vorhandenen Platten werden nicht betrachtet, da sie hauptsächlich für das Betriebssystem genutzt werden. Somit sind für einen Application-Server allein die Parameter der CPUs entscheidend:

- *Performance-Index:* Leistungsmaß für die CPUs in SAPS (für SAPS-Definition siehe Kapitel 02 „Monitoring und Benchmarking für das R/3-Performance-Engineering“)
- *CPU-Anzahl:* Anzahl der CPUs
- *Skalierungsfaktoren:* Faktoren, die das Ansteigen der Leistung beim Multi-Prozessorbetrieb beschreiben

Die Eingabe von Application-Servern wird im WLPSizer durch den in Abbildung 03-16 dargestellten Dialog ermöglicht. Abgefragt werden der Name des Servers und der CPU-Typ. Die CPU kann mittels der Bibliotheken ausgewählt werden, womit der Performance-Index, die Anzahl der CPUs und die Skalierungsfaktoren bestimmt sind.

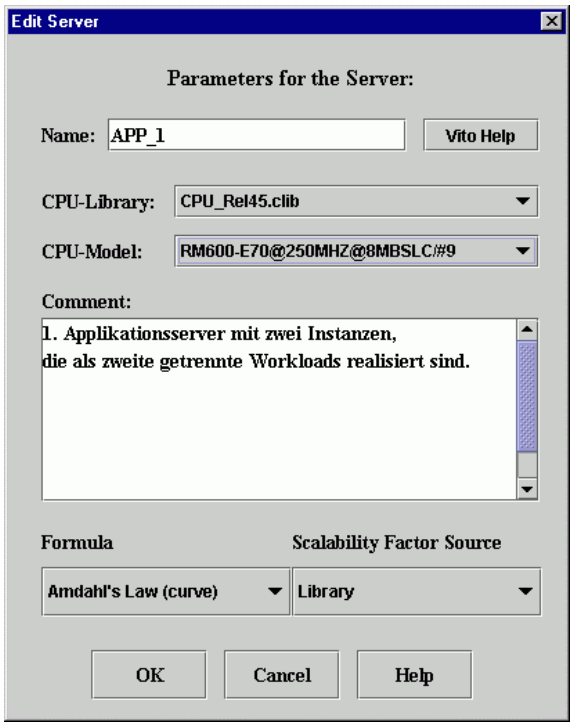


Abbildung 03-16: Application-Server-Dialog

Die Skalierungsfaktoren der Prozessoren werden im WLPSizer berechnet. Es kann unter Formula die zu verwendende Formel ausgewählt werden. Es stehen die Berechnungsformeln von Amdahl und Gustafson (siehe [5]) zur Verfügung. Zur Berechnung der Faktoren wird der Skalierungsfaktor des ausgewählten CPU-Typs, womit auch die Anzahl der Prozessoren festgelegt wird, benötigt. Unter Scalability Factor Source wird eingestellt, ob der zum CPU-Typ zugehörige Skalierungsfaktor der Bibliothek entnommen oder neu eingegeben wird. Wurde ein CPU-Typ mit mindestens drei Prozessoren ausgewählt (Anzahl N), so werden für die Fälle 2, ..., N-1 die Skalierungsfaktoren mit Hilfe des WLPSizers interpoliert.

(e) Database-Server und IO-System

Der Database-Server ist einer Erweiterung des Application-Servers, so dass die bereits für den Application-Server beschriebenen Eingabeparameter für den Database-Server ebenfalls eingegeben werden, d.h. Name, CPU und Skalierungsfaktoren. Die Erweiterung des Database-Servers umfasst die folgenden Parameter:

- *DB-Hit-Ratio [%]*: Anteil an Datenbankzugriffen, die aus dem Cache der Datenbank bedient werden können und somit keine IO-Aktivität am Speichersystem benötigen
- *Seek-Time [s]*: mittlerer Zeitbedarf zur Positionierung der Schreib-Lese-Köpfe der Platte
- *Rotationsgeschwindigkeit [RPM[28]]:* Rotationsgeschwindigkeit der Platte
- *Blocksize [Byte]*: Blockgröße der Platte
- *Transferrate [MB/s]*: mittlere Geschwindigkeit der Platte für Lese- und Schreibzugriffe
- *Cache-Hit-Ratio [%]:* Anteil an IO-Zugriffen, die aus dem Cache der Platte bedient werden können
- *Controller-Time [s]:* mittlerer Zeitbedarf der Platte, der für einen IO-Request am Controller benötigt wird. Es sind enthalten die Zeit zur Überprüfung des Caches und die Zeit, einen Block in den Cache zu lesen bzw. aus dem Cache heraus zu schreiben.
- *Access-Distribution:* Verteilung der Zugriffe auf die Platten. Es kann zwischen Gleichverteilung oder Sägezahnverteilung ausgewählt werden. Bei letzterer verteilen sich die Zugriffe asymmetrisch auf die Platten nach folgender Formel:
  $\text{Access}(i) = i / \Sigma_{j=1..N} j = 2i / (N^2 + N)$
  mit Access (i): Anteil an IO-Zugriffen, die auf die i-te Platte erfolgen
  N: Anzahl der Platten

Es ergibt sich bei zum Beispiel drei Platten die Zugriffsverteilung 1/6, 1/3 und 1/2. Die zugehörige Sägezahnverteilung hat die in Abbildung 03-17 illustrierte Gestalt.

[28] RPM = Rounds Per Minute

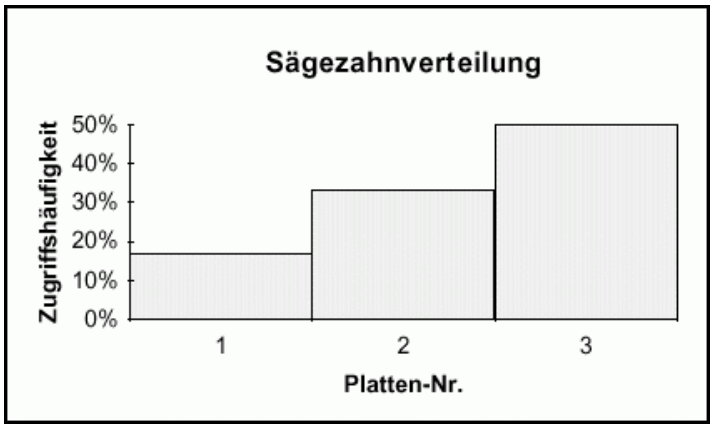


Abbildung 03-17: Asymmetrische Sägezahnverteilung auf die Platten

Zur Eingabe eines Database-Servers wird der in Abbildung 03-18 dargestellte Dialog verwendet. Es werden die bereits beschriebenen Angaben für die CPUs, Datenbank und das IO-System abgefragt. Für die CPUs kann die jeweilige CPU-Bibliothek und der entsprechende Eintrag ausgewählt werden. Für die Datenbank wird zusätzlich die DB-Hit-Ratio abgefragt.

Wie bereits für die CPUs beschrieben, wird auch das IO-System mittels Bibliotheken ausgewählt (Disk-Library). Unter Disk-Type kann das den Anforderungen entsprechende IO-System ausgewählt werden. Mit Ausnahme der Access-Distribution und der Platten-Anzahl werden alle Parameter für das IO-System durch die Bibliothek geliefert. Die beiden fehlenden Parameter werden im DB-Server-Dialog spezifiziert. Die IO-Bibliothek liefert Standard-Werte für die Blocksize und die Cache-Hit-Ratio der Platten, die jedoch im Dialog modifiziert werden können.

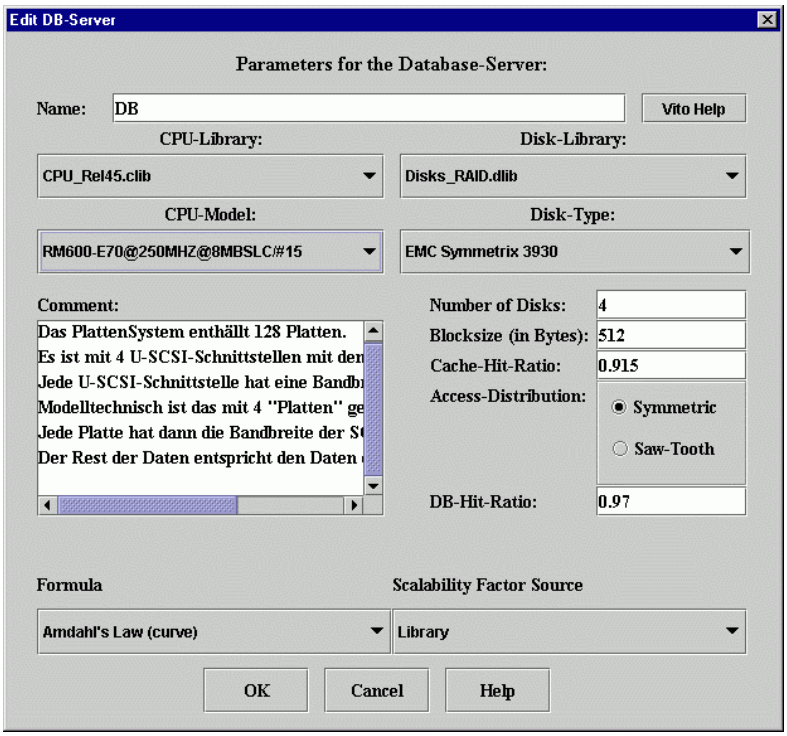


Abbildung 03-18: Database-Server-Dialog

(f) Netze

Netze werden im WLPSizer zur Verbindung der Clientgroups mit den Application-Servern und der Application-Server mit dem Database-Server benötigt. Es können auch Netze miteinander verbunden werden. Die Parameter eines Netzes sind wie folgt:

- *Length [m]:* Länge des Netzwerks
- *Signalgeschwindigkeit [m/s]:* Signalgeschwindigkeit des Mediums
- *Bandbreite [Mbit/s]:* Bandbreite des Mediums
- *Packetlength [Byte]:* mittlere Paketgröße
- *Protocol:* Zugangsprotokoll
- *Overhead per Packet [Byte]:* Overhead pro Paket

@ *Maximum Packet Length [Byte]:* maximale Paketgröße

@ *Connections:* an das Netz angeschlossene Server, Clientgroups und Netze

Zur Eingabe von Netzen wird der in Abbildung 03-19 dargestellte Dialog verwendet. Es können der Name, die Länge des Mediums und die mittlere Paketgröße angegeben werden. Das Protokoll wird der Bibliothek entnommen. Die Bibliothek liefert die Parameter für die Signalgeschwindigkeit, Bandbreite, Overhead und die maximale Paketlänge, wobei die letzten beiden Parameter Standardwerte im Netz-Dialog darstellen, die nachträglich modifiziert werden können.

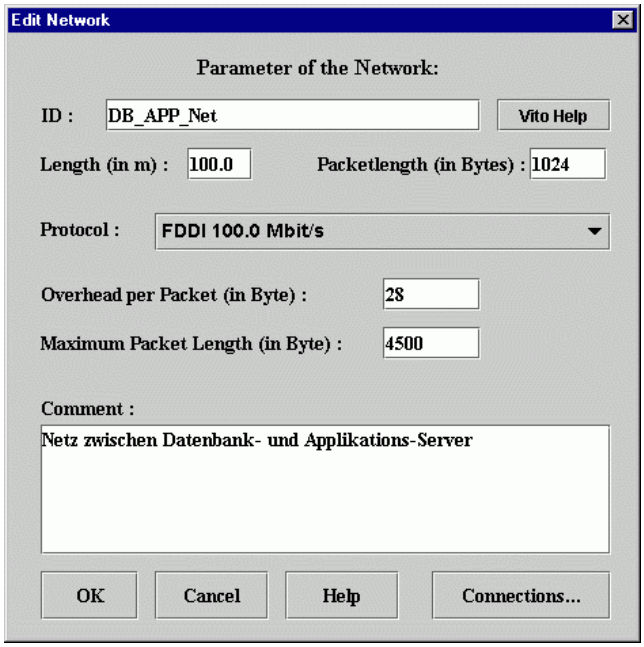


Abbildung 03-19: Netz-Dialog

Die Schaltfläche Connections öffnet einen Dialog zur Spezifikation der an das Netz anzuschließenden Konfigurationselemente (siehe Abbildung 03-20).

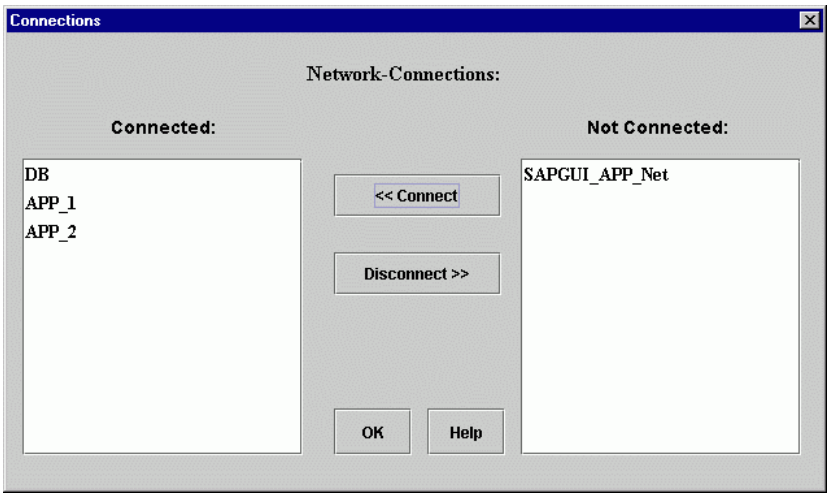


Abbildung 03-20: Connections-Dialog für ein Netz

Der Verbindungsdialog besteht im Wesentlichen aus zwei Listen. In der ersten Liste sind die Server und Netze der Konfiguration aufgeführt, die mit diesem Netz verbunden sind (Connected). In der zweiten Liste sind diejenigen Komponenten enthalten, die nicht mit dem Netz verbunden sind (Not Connected). Mittels der Schaltflächen Connect bzw. Disconnect können die jeweils ausgewählten Objekte mit dem Netz verbunden bzw. von ihm getrennt werden. Es ist hierbei zu beachten, dass für jeden Server eine Verbindung zum Database-Server existieren muss.

(g) Settings einer Konfiguration

Die Settings einer Konfiguration sind Parameter, die zum Teil anhand der Messdaten berechnet werden müssen. Der zugehörige Dialog wird in Abbildung 03-21 dargestellt.

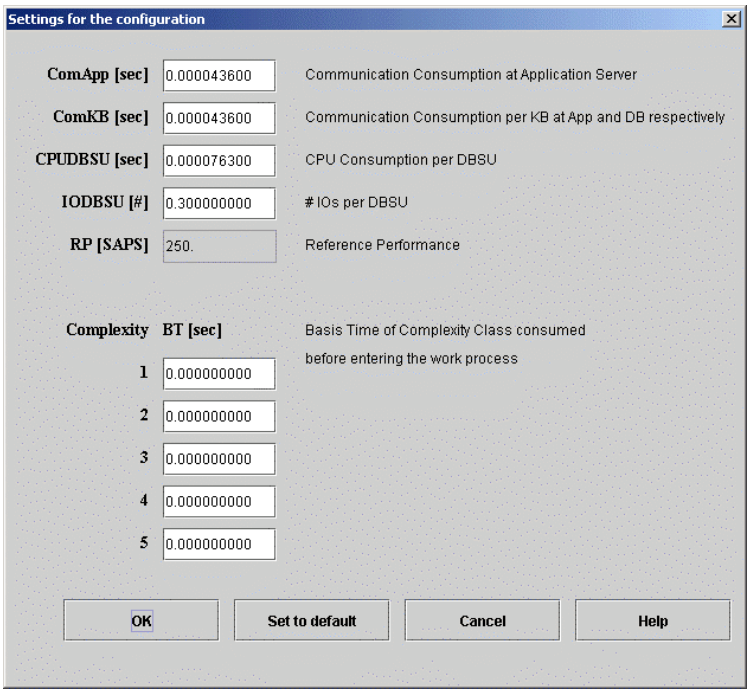


Abbildung 03-21: Settings einer Konfiguration

Die folgenden Parameter können spezifiziert werden:

Parameter	Einheit	Beschreibung
ComApp	Sek.	Basis-CPU-Bedarf für die Kommunikation am Application-Server
ComKB	Sek.	CPU-Bedarf für die Kommunikation am Application- bzw. DB-Server pro KB
CPUDBSU	Sek.	CPU-Bedarf pro DBSU am DB-Server
IODBSU	Anzahl	Anzahl der Ios pro DBSU
RP	SAPS	Referenzleistung zur Normierung der Workloads (= 250 SAPS)
BT 1..5	Sek.	Basis-CPU-Bedarf für Dialogschritte der Komplexitätsklassen 1..5, wie z.B. CPU-Zeit am Dispatcher

Tabelle 03-6: Settings-Parameter

Die Referenzleistung kann nicht modifiziert werden. Es wird für alle Konfigurationen von einer festen Referenzleistung (250 SAPS) ausgegangen. Die Konstante CPUDBSU kann anhand der Messdaten berechnet werden (siehe Abschnitt 03.06 „Erweiterte Funktionen und Vorgehensweisen zur Modellierung“).

03.04 Modell-Evaluation

Die Modell-Evaluation im WLPSizer beinhaltet die Erzeugung und Berechnung des Modells. Die definierte Konfiguration wird hierbei mit einer Build-Funktion in ein Warteschlangennetz überführt. Bei Inkonsistenzen in der Konfiguration wird eine Fehlermeldung angezeigt. Inkonsistenzen betreffen hierbei die Topologie der Konfiguration, wie zum Beispiel, dass ein Application-Server nicht mit dem DB-Server verbunden ist.

Mit einer Evaluate-Funktion kann das Warteschlangennetz mittels TOTO gelöst werden. Es werden die Daten zum Warteschlangenlöser TOTO gesendet, der das Modell löst und die Ergebnisse zurück zum WLPSizer sendet. Der WLPSizer stellt die Ergebnisse dann mit Hilfe verschiedener Sichten dar (siehe folgenden Abschnitt).

03.05 Ergebnisse

Die mit der Modell-Evaluation ermittelten Ergebnisse werden im WLPSizer in verschiedenen Sichten dargestellt und mittels drei verschiedener Dienstgüten (good, moderate, bad) bewertet. Nachfolgend werden das Dienstgütekonzept und die verschiedenen Ergebnis-Sichten des WLPSizers beschrieben. Abschließend wird die Report-Funktion des WLPSizers dargestellt, mit der die Ergebnisse und Konfigurationsbeschreibungen zusammengefasst und exportiert werden können.

(a) Dienstgüte-Management

In Abhängigkeit der Komplexitätsklassen werden drei Dienstgüteklassen definiert. Die Definition der Dientgüteklassen basiert auf der Antwortzeit der Dialogschritte, die in Abhängigkeit der Komplexitätsklasse durch Schwellenwerte spezifiziert werden kann (siehe Tabelle 03-7).

Dienstgüteklasse	**Kompl.[29] 1**	**Kompl. 2**	**Kompl. 3**	**Kompl. 4**	**Kompl. 5**
Good	<= 200	<= 1000	<= (#DBSU)[30]	<= (#DBSU)	<= (#DBSU)
Moderate	<= 800	<= 4000	<= 4*(#DBSU)	<= 4*(#DBSU)	<= 4*(#DBSU)
Bad	> 800	> 4000	> 4*(#DBSU)	> 4*(#DBSU)	> 4*(#DBSU)

Tabelle 03-7: Dienstgüteklassen

[29] Kompl. = Komplexitätsklasse
[30] (#DBSU) = 1 ms pro DB-Service-Unit

Beispiel:

- Ein Dialogschritt der Komplexitätsklasse 1 wird somit der Dienstgüteklasse good (moderate) zugeordnet, wenn seine Antwortzeit (RespTime) unter 200 (800) ms liegt. Übersteigt die Antwortzeit 800 ms, besitzt der Dialogschritt die Dienstgüte bad.
- Ein Dialogschritt der Komplexitätsklasse 3 mit 1500 DBSU besitzt als Schwellenwert 1500 ms. Die Antwortzeit für Dienstgüte good (moderate) muss demnach unter 1,5 (6) Sekunden liegen. Über 6 Sekunden wird der Dialogschritt der Dienstgüte bad zugeordnet.

(b) Ergebnis-Sichten

Die Ergebnisse der Modell-Evaluation werden im WLPSizer in vier verschiedenen Sichten dargestellt, die nachfolgend beschrieben werden. Treten bei der Evaluation des Modells Fehler auf, so können diese in allen vier Sichten mittels der Schaltfläche Details angezeigt werden. Die Ergebnisse der Server-, Workload- und Utilization-View werden in gleicher Form wie die Workload-Profile gruppiert nach Tasktyp und Komplexitätsklasse. Es werden für jede Klasse (Kombination aus Tasktyp und Komplexitätsklasse) Mittelwerte (Verbrauch pro Dialogschritt) angegeben.

Server-View

Die Server-View stellt die Modell-Ergebnisse aus Sicht eines bestimmten Servers (DB oder Application-Server) dar. Es werden die folgenden Daten angezeigt:

- Name der Konfiguration
- Name des Servers
- CPU-Auslastung des Servers laut Modellrechnung

Zusätzlich sind Analyseergebnisse über die auf diesem Server verarbeiteten Dialogschritte in einer Tabelle zusammengestellt (siehe Abbildung 03-22).

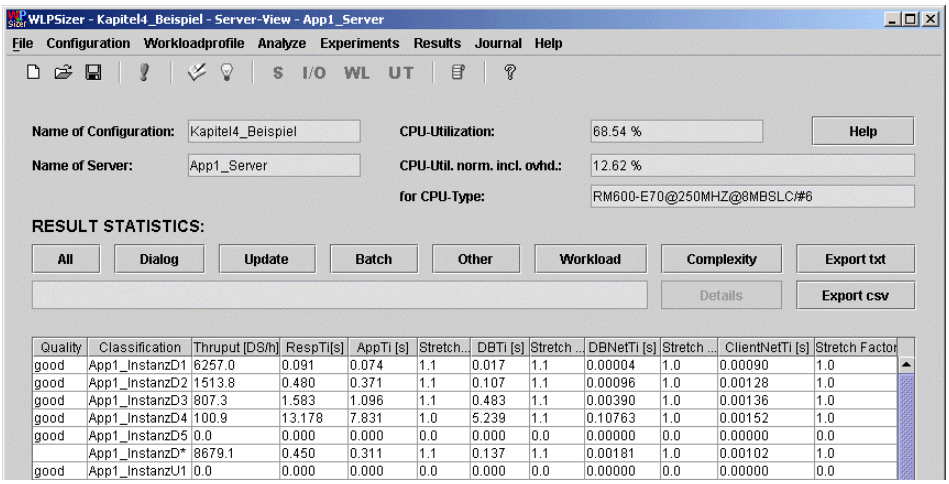

WLPSizer - Kapitel4_Beispiel - Server-View - App1_Server
File Configuration Workloadprofile Analyze Experiments Results Journal Help
S I/O WL UT
Name of Configuration: Kapitel4_Beispiel
Name of Server: App1_Server
CPU-Utilization: 68.54 %
CPU-Util. norm. incl. ovhd.: 12.62 %
for CPU-Type: RM600-E70@250MHZ@8MBSLC/#6
Help
RESULT STATISTICS:
All Dialog Update Batch Other Workload Complexity Export txt
Details Export csv
Quality Classification Thruput [DS/h] RespTi[s] AppTi [s] Stretch.. DBTi [s] Stretch .. DBNetTi [s] Stretch .. ClientNetTi [s] Stretch Factor
good App1_InstanzD1 6257.0 0.091 0.074 1.1 0.017 1.1 0.00004 1.0 0.00090 1.0
good App1_InstanzD2 1513.8 0.480 0.371 1.1 0.107 1.1 0.00096 1.0 0.00128 1.0
good App1_InstanzD3 807.3 1.583 1.096 1.1 0.483 1.1 0.00390 1.0 0.00136 1.0
good App1_InstanzD4 100.9 13.178 7.831 1.0 5.239 1.1 0.10763 1.0 0.00152 1.0
good App1_InstanzD5 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0
App1_InstanzD* 8679.1 0.450 0.311 1.1 0.137 1.1 0.00181 1.0 0.00102 1.0
good App1_InstanzU1 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0

Abbildung 03-22: Server-View

Es werden für die Dialogschritte die folgenden Daten angezeigt:

Parameter	**Einheit**	**Beschreibung**
Quality	---	Dienstgüte (good, moderate, bad)
Classification	---	Name der Workload, gefolgt von Tasktyp und Komplexitätsklasse des Dialogschritts
Thruput	DS/h	Durchsatz
RespTi	Sek.	Antwortzeit des Dialogschritts (ClientNetTi+AppTi+DBNetTi+DBTi)
AppTi	Sek.	Zeit am Application-Server inkl. Wartezeiten (= 0 bei DB-Server)
Stretch Factor (Appl.-Server)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung am Application-Server
DBTi	Sek.	Zeit am DB-Server inkl. der IO-Zeiten und Wartezeiten
Stretch Factor (DB-Server)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung am DB-Server
DBNetTi	Sek.	Zeit auf dem Netz bzw. den Netzen zwischen Application- und DB-Server (inkl. Wartezeiten)
Stretch Factor (DB-Netz)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung am DB-Netz
ClientNetTi	Sek.	Zeit auf dem Netz bzw. den Netzen zwischen Application-Server und Clientgroups
Stretch Factor (Client-Net)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung am Client-Netz

Tabelle 03-8: Parameter der Server-View

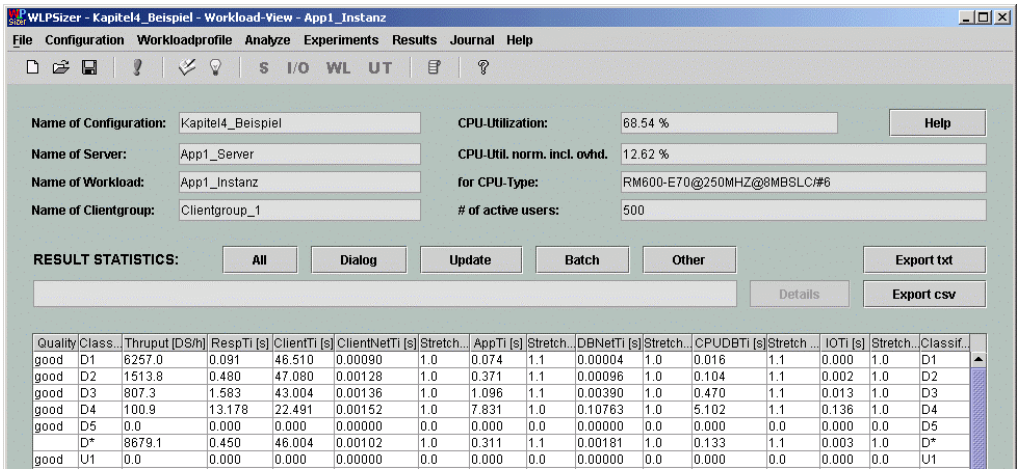
In der tabellarischen Ergebnisdarstellung wird das Symbol „*“ als Platzhalter für den Tasktyp bzw. die Komplexitätsklasse verwendet. Die Werte dieser Zeile sind eine Zusammenfassung aller Dialogschritte mit fester Komplexitätsklasse und beliebigem Tasktyp bzw. festem Tasktyp und beliebiger Komplexitätklasse. Dabei wird der Durchsatz als Summe der Durchsätze der einzelnen Dialogschritte berechnet. Alle weiteren Werte sind gewichtete Mittelwerte, die auf Grundlage des Durchsatzes bestimmt werden.

Workload-View

Die Workload-View liefert die Modell-Ergebnisse aus Sicht einer ausgewählten Workload. Es werden die folgenden Daten dargestellt:

- Name der Konfiguration
- Name der Workload
- Name der Clientgroup (Quelle der Workload)
- Anzahl der Terminals dieser Clientgroup
- Name des Servers (Ziel der Workload)
- CPU-Auslastung des Servers

In einer Tabelle werden die berechneten Zeiten für die Dialogschritte dieser Workload zusammengestellt (siehe Abbildung 03-23).



Quality	Class...	Thruput [DS/h]	RespTi [s]	ClientTi [s]	ClientNetTi [s]	Stretch...	AppTi [s]	Stretch...	DBNetTi [s]	Stretch...	CPUDBTi [s]	Stretch ...	IOTi [s]	Stretch...	Classif...
good	D1	6257.0	0.091	46.510	0.00090	1.0	0.074	1.1	0.00004	1.0	0.016	1.1	0.000	1.0	D1
good	D2	1513.8	0.480	47.080	0.00128	1.0	0.371	1.1	0.00096	1.0	0.104	1.1	0.002	1.0	D2
good	D3	807.3	1.583	43.004	0.00136	1.0	1.096	1.1	0.00390	1.0	0.470	1.1	0.013	1.0	D3
good	D4	100.9	13.178	22.491	0.00152	1.0	7.831	1.0	0.10763	1.0	5.102	1.1	0.136	1.0	D4
good	D5	0.0	0.000	0.000	0.00000	0.0	0.000	0.0	0.00000	0.0	0.000	0.0	0.000	0.0	D5
	D*	8679.1	0.450	46.004	0.00102	1.0	0.311	1.1	0.00181	1.0	0.133	1.1	0.003	1.0	D*
good	U1	0.0	0.000	0.000	0.00000	0.0	0.000	0.0	0.00000	0.0	0.000	0.0	0.000	0.0	U1

Abbildung 03-23: Workload-View

In der Ergebnistabelle sind die folgenden Werte enthalten:

Parameter	Einheit	Beschreibung
Quality	---	Dienstgüte (good, moderate, bad)
Classification	---	Tasktyp und Komplexitätsklasse des Dialogschritts
Thruput	DS/h	Durchsatz
RespTi	Sek.	Antwortzeit des Dialogschritts (ClientNetTi+AppTi+DBNetTi+CPUDBTi+IOTi)
ClientTi	Sek.	Zeit des Benutzers zur Bearbeitung des Dialogschritts (inklusive Denkzeit)
ClientNetTi	Sek.	Zeit auf dem Netz bzw. den Netzen zwischen Application-Server und Clientgroups (inkl. Wartezeiten)
Stretch Factor (Client-Net)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung auf dem Client-Net
AppTi	Sek.	Zeit am Application-Server inkl. Wartezeiten (= 0 bei DB-Server)
Stretch Factor (Appl.-Server)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung am Application-Server
DBNetTi	Sek.	Zeit auf dem Netz bzw. Netzen zwischen Application- und DB-Server (inklusive Wartezeiten)
Stretch Factor (DB-Netz)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung am DB-Netz
CPUDBTi	Sek.	CPU-Zeit am DB-Server (inklusive Wartezeiten)
Stretch Factor (DB-CPU)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung an DB-CPU
IOTi	Sek.	IO-Zeit des Dialogschritts am DB-Srver (inklusive Wartezeiten
Stretch Factor (IO-System)	---	Dehnfaktor: Verhältnis von Verweilzeit zur Bedienzeitanforderung am IO-System

Tabelle 03-9: Parameter der Workload-View

Utilization-View

In der Utilization-View werden die durch Dialogschritte verursachten Auslastungen der Systemkomponenten dargestellt. Diese Darstellung wird jeweils für eine vorher ausgewählte Workload erstellt. Es werden mit Ausnahme der Ergebnistabelle die gleichen Daten, die bereits für die Workload-View beschrieben wurden, angezeigt (siehe Abbildung 03-24).

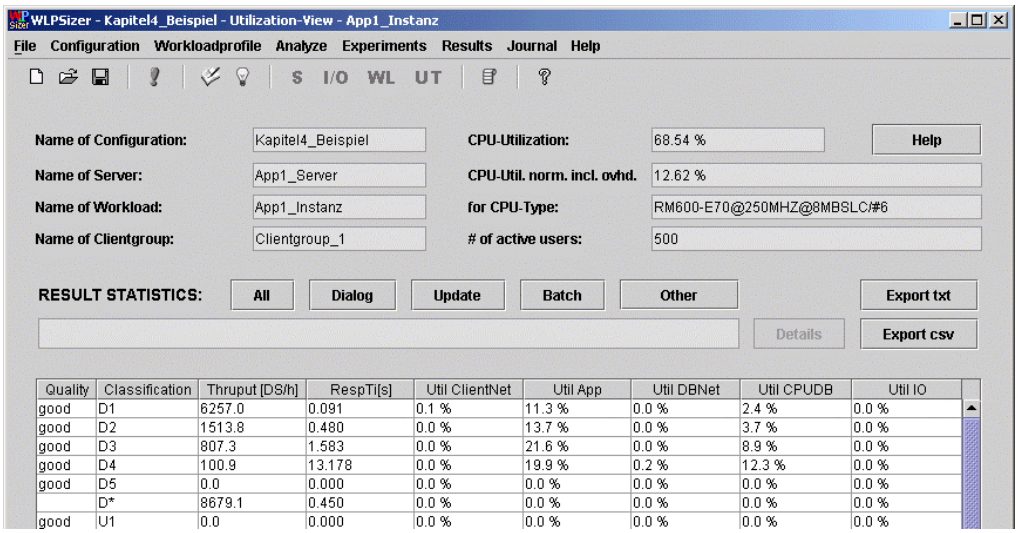

WLPSizer - Kapitel4_Beispiel - Utilization-View - App1_Instanz
File Configuration Workloadprofile Analyze Experiments Results Journal Help
S I/O WL UT
Name of Configuration: Kapitel4_Beispiel
Name of Server: App1_Server
Name of Workload: App1_Instanz
Name of Clientgroup: Clientgroup_1
CPU-Utilization: 68.54 %
CPU-Util. norm. incl. ovhd. 12.62 %
for CPU-Type: RM600-E70@250MHZ@8MBSLC/#6
# of active users: 500
Help
RESULT STATISTICS:
All
Dialog
Update
Batch
Other
Export txt
Details
Export csv
Quality Classification Thruput [DS/h] RespTi[s] Util ClientNet Util App Util DBNet Util CPUDB Util IO
good D1 6257.0 0.091 0.1 % 11.3 % 0.0 % 2.4 % 0.0 %
good D2 1513.8 0.480 0.0 % 13.7 % 0.0 % 3.7 % 0.0 %
good D3 807.3 1.583 0.0 % 21.6 % 0.0 % 8.9 % 0.0 %
good D4 100.9 13.178 0.0 % 19.9 % 0.2 % 12.3 % 0.0 %
good D5 0.0 0.000 0.0 % 0.0 % 0.0 % 0.0 % 0.0 %
D* 8679.1 0.450 0.0 % 0.0 % 0.0 % 0.0 % 0.0 %
good U1 0.0 0.000 0.0 % 0.0 % 0.0 % 0.0 % 0.0 %

Abbildung 03-24: Utilization-View

Die Ergebnistabelle enthält die folgenden Werte:

Parameter	Einheit	Beschreibung
Quality	---	Dienstgüte (good, moderate, bad)
Classification	---	Tasktyp und Komplexitätsklasse des Dialogschritts
Thruput	DS/h	Systemdurchsatz für den Dialogschritt
RespTi	Sek.	Antwortzeit des Dialogschritts (AppTi+NetTi+CPUDBTi+IOTi)
Util ClientNet	%	Auslastung des Client-Netzes (bei mehreren Netzen wird die höchste Auslastung genommen)
Util App	%	Auslastung am Application-Server
Util DBNet	%	Auslastung des Netzes zwischen Application und DB-Server
Util CPUDB	%	Auslastung an der CPU des DB-Servers
Util IO	%	Auslastung des IO-Systems

Tabelle 03-10: Parameter der Utilization-View

IO-View

Die IO-View liefert die Auslastung des IO-Systems vom DB-Server und die folgenden Angaben:

@ Name der Konfiguration

@ Name des DB-Servers

@ CPU-Auslastung am DB-Server

Die IO-View liefert für das IO-System die Auslastung der einzelnen Platten in Prozent (siehe Abbildung 03-25).

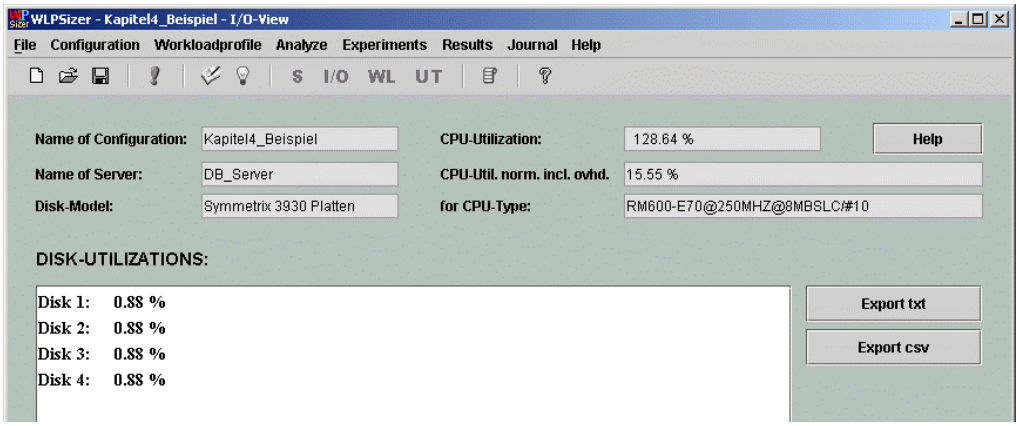


Abbildung 03-25: IO-View

Reports

Im WLPSizer können Berichte über die erstellten Konfigurationen generiert werden. Es können bereits vorkonfigurierte Berichte oder einzelne Abschnitte der vorhandenen Ergebnisse und Konfigurationsbeschreibungen aufgerufen werden (siehe Abbildung 03-26).

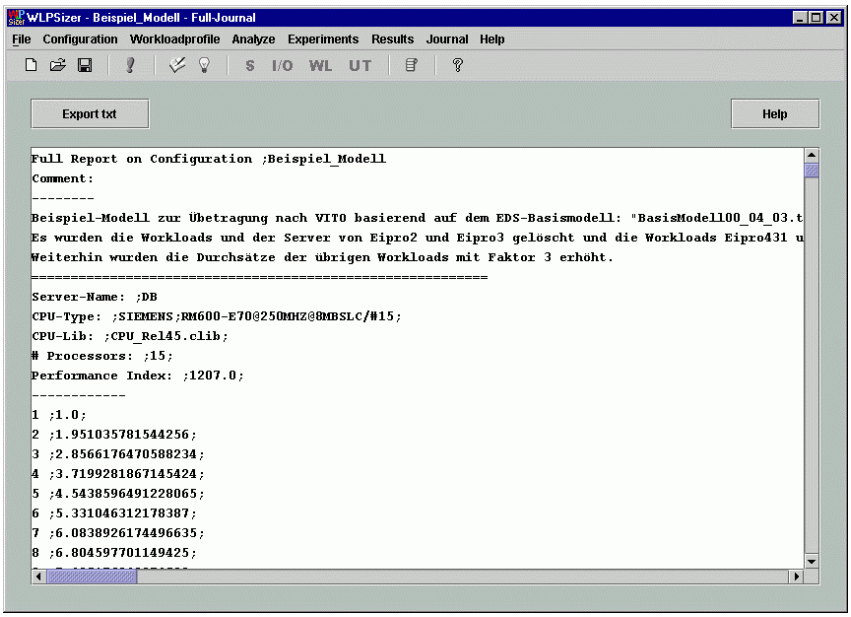


Abbildung 03-26: Darstellung von Reports

Es kann zum Beispiel der Full Report als vorkonfigurierter Bericht ausgewählt werden, der alle zur Verfügung stehenden Informationen bez. einer Konfiguration enthält:

- Konfigurationsbeschreibung
- Spezifikation der einzelnen Konfigurations-Komponenten
- Workload-Profile
- Warteschlangenmodell
- Ergebnisse des Warteschlangenlösers
- Ergebnissichten des WLPSizers
- Dienstgüte-Definitionen

Die Daten werden im WLPSizer dargestellt und können als Text-Datei exportiert werden. Mit Hilfe von MS Excel können die Daten weiterverarbeitet werden.

03.06 Erweiterte Funktionen und Vorgehensweisen zur Modellierung

Die bislang dargestellten Eigenschaften des WLPSizers können als Grundeigenschaften zusammengefasst werden. Im folgenden Abschnitt werden ausgehend von diesen Grundeigenschaften die erweiterten Funktionen des WLPSizers beschrieben. Es werden hierbei zwei weitere Typen von Workloads und die Berechnung von Settings-Parametern als weitere Modell-Eingabe-Parameter vorgestellt. Des Weiteren werden typische Prognose-Szenarien, die mit dem WLPSizer betrachtet werden können, dargestellt.

(a) Unterstützung verschiedener Workload-Typen

Im WLPSizer werden mittels Workloads die gemessenen bzw. definierten R/3-Lasten in das Modell eingegeben. Es werden hierfür drei verschiedene Workload-Typen bereitgestellt:

@ Standard: Die Workload besitzt eine Quelle (Clientgroup) und ein Ziel (Server). Der Pfad zwischen Quelle und Ziel wird automatisch vom WLPSizer festgelegt.

@ Background: Die Workload besitzt ein Ziel, das eine beliebige Komponente des Modells sein kann, wie z.B. Netz, IO-System oder CPUs. Der Workload wird automatisch eine Quelle zugewiesen, die im WLPSizer als nicht sichtbare Modell-Komponente existiert.

@ Custom: Die Workload besitzt eine Quelle (Clientgroup). Der Pfad für die Workload bzw. die zu besuchenden Komponenten des Modells ist frei definierbar. Für alle Komponenten können Besuchszahlen angegeben werden, die festlegen, wie oft sie von der Workload besucht werden sollen.

Bislang wurde nur der Workload-Typ Standard beschrieben. In den folgenden Abschnitten werden die beiden Typen Background und Custom dargestellt.

Background-Workload

Die Motivation für Hintergrundlasten besteht in der gezielten Belastung einzelner Modellkomponenten. Für die Hintergrundlast kann somit im WLPSizer-Dialog die zu belastende Komponente (Destination) festgelegt werden, wobei dies eine beliebige Station des Modell sein kann (siehe Abbildung 03-27).

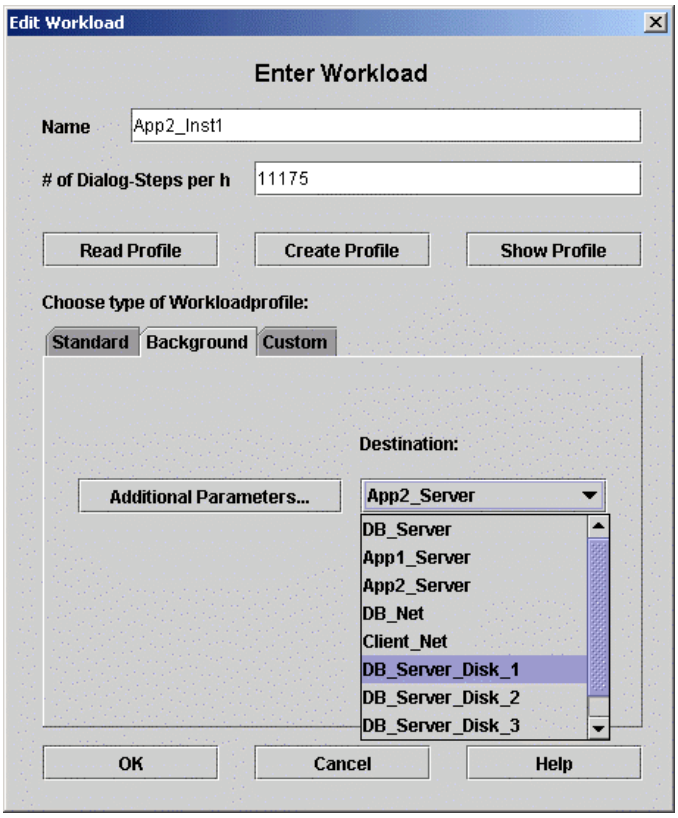


Abbildung 03-27: Hintergrundlast-Dialog

Anschließend können über die Schaltfläche Additional Parameters weitere Parameter für die Hintergrundlast spezifiziert werden (siehe Abbildung 03-28). Es kann zum einen die Anzahl der User, die die Hintergrundlast verursachen, eingegeben werden. Zum anderen gibt es die Möglichkeit, die Durchsatzsteuerung zu aktivieren (Thinktime Output Parameter), so dass die Denkzeiten der User berechnet und die eingegebenen Durchsätze erreicht werden, bzw. die Durchsatzsteuerung zu deaktivieren, so dass Denkzeiten für die angegebenen User eingegeben werden können.

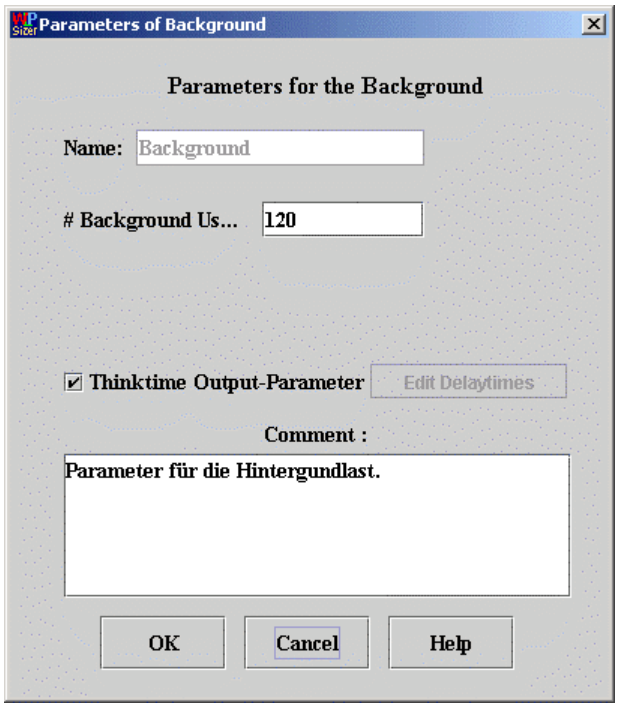


Abbildung 03-28: Parameter für die Hintergrundlast

Der in Abbildung 03-28 dargestellte Dialog entspricht zum größten Teil dem der Clientgroups, da für jede Hintergrundlast eine fiktive Clientgroup im Modell erstellt wird, die aber im WLPSizer nicht sichtbar ist. Diese fiktive Clientgroup wird dann als Quelle der Last verwendet.

Custom Workload

Die bislang beschriebenen Workload-Typen besitzen automatisch festgelegte Pfade, die die zu besuchenden Komponenten der Konfiguration vorgeben, so dass Quelle und Ziel der Workload erreicht werden. Für Custom-Workloads hingegen ist der Pfad frei definierbar. Für jede Komponente des Modells, mit Ausnahme der Clientgroups, kann die Anzahl der Besuche selbst eingegeben werden. Es muss lediglich die Quelle (Clientgroup) der Workload angegeben werden (siehe Abbildung 03-29), die die Last erzeugt.

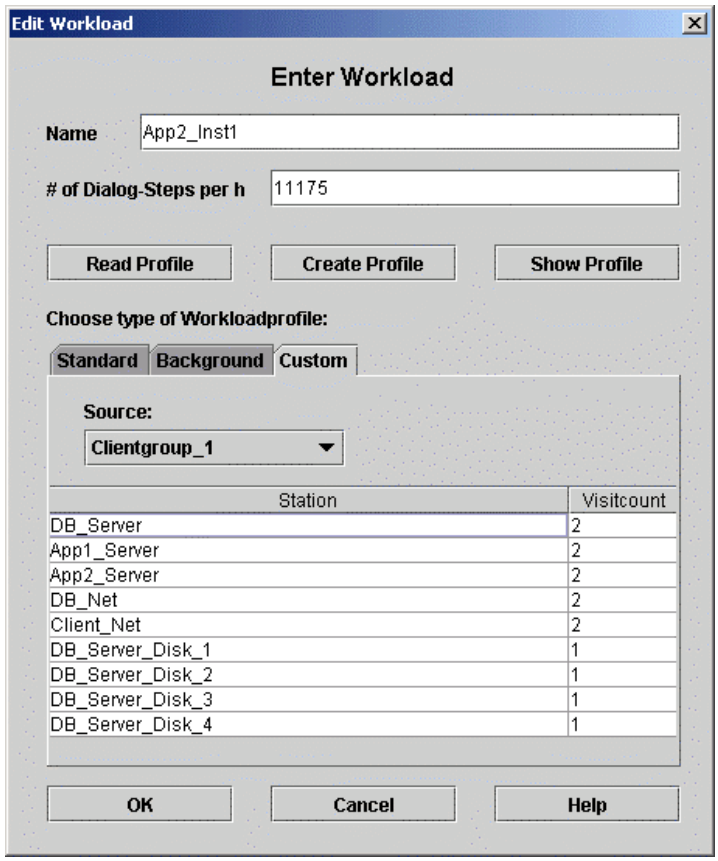


Abbildung 03-29: User defined-Workload

(b) Spezifikation der Settings-Parameter

Die Settings-Parameter einer Konfiguration werden zur Kalibrierung der Modelle verwendet. In diesem Abschnitt werden Berechnungen für einzelne Parameter der Settings vorgestellt.

Berechnung der CPUDBSU-Konstante

Die CPUDBSU-Konstante beschreibt den CPU-Verbrauch am DB-Server pro DBSU (Database-Service-Unit). Die Auslastung des DB-Servers pro Prozessor ergibt sich im Modell wie folgt:

Auslastung DB-Server = U(DB) =

$$\frac{\alpha \bullet \left(\sum_{i \in WL} \sum_{j=1}^{20} D_i \bullet ratio_{ji} \bullet DBSU_{ji} \bullet CPUDBSU + \sum_{i \in WL_{DB}} \sum_{j=1}^{20} D_i \bullet ratio_{ji} \bullet CPUTI_{ji} \right)}{P \bullet T}$$

Parameter	**Beschreibung**
U(DB)	Auslastung des DB-Servers pro Prozessor
WL	Menge aller Workloads des Modells
D_i	Durchsatz der i-ten Workload
WL_{DB}	Menge aller Workloads mit dem Ziel DB-Server
$ratio_{ji}$	Spalte ratio der j-ten Zeile der i-ten Workload
$CPUTI_{ji}$	Spalte CPUTI der j-ten Zeile der i-ten Workload (ms)
T	Zeitintervall (ms)
P	Anzahl der Prozessoren des DB-Servers
α	α = RP / (L/Skal) Faktor der Workload-Normierung mit: RP = Referenzleistung (250 SAPS) L = SAPS-Leistung des DB-Servers Skal = Skalierungsfaktor für aktuelle Anzahl der Prozessoren

Tabelle 03-11: Parameter zur Auslastungsberechnung des DB-Servers

Für die Berechnung der DB-Server-Auslastung werden die im Zähler stehenden CPU-Zeitverbräuche durch das Zeitintervall dividiert. Im WLPSizer wird im Stundenmittel gerechnet. Daher kann T gleich einer Stunde gesetzt werden. Die erste Doppelsumme des Zählers berechnet aus den DBSU-Verbräuchen aller Workloads die konsumierten CPU-Zeiten, indem die verbrauchten DBSUs mit der Konstante CPUDBSU multipliziert werden. Die zweite Doppelsumme berechnet die CPU-Zeiten der auf dem DB-Server verarbeiteten Workloads, d.h. Workloads mit dem DB-Server als Ziel. Die jeweils zweite Summe der Doppelsummen durchläuft die einzelnen Zeilen der Workloads und besitzt somit als obere Grenze j = 20, da eine Workload aus 20 Zeilen besteht. Die in der Doppelsumme verwendete zweite Summe kann durch die in den Workloads stehende **-Zeile ersetzt werden, da die **-Zeile den gewichteten Mittelwert repräsentiert. Es gilt:

$$\sum_{j=1}^{20} D \bullet ratio_j \bullet DBSU_j = D \bullet DBSU^{**}$$

$$\sum_{j=1}^{20} D \bullet ratio_j \bullet CPUTI_j = D \bullet CPUTI^{**}$$

Mittels Umformungen erhält man die folgende Berechnungsformel für den Parameter CPUDBSU:

CPUDBSU=

$$\frac{U(DB) \bullet P \bullet T - \left(\sum_{i \in WL_{DB}} \alpha \bullet CPUTI_i^{**} \bullet D_i\right)}{\left(\sum_{i \in WL} D_i \bullet DBSU_i^{**}\right) \bullet \alpha}$$

Parameter	Beschreibung
U(DB)	Auslastung des DB-Servers pro Prozessor
WL	Menge aller Workloads des Modells
D_i	Durchsatz der i-ten Workload
WL_{DB}	Menge aller Workloads mit dem Ziel DB-Server
$DBSU^{**}$	Spalte DBSU der **-Zeile der i-ten Workload
$CPUTI_{ji}$	Spalte CPUTI der **-Zeile der i-ten Workload (ms)
T	Zeitintervall (ms)
α	α = RP / (L/Skal) Faktor der Workload-Normierung mit: RP = Referenzleistung (250 SAPS) L = SAPS-Leistung des DB-Servers Skal = Skalierungsfaktor für aktuelle Anzahl der Prozessoren

Tabelle 03-12: Parameter zur Berechnung des CPUDBSU-Parameters

Der berechnete CPUDBSU-Parameter kann dann in die Settings der Konfiguration eingetragen werden. Die resultierende Auslastung der Datenbank entspricht dann U(DB). Die Konstante CPUDBSU ermöglicht somit die Kalibrierung des DB-Servers, so dass dieser im Modell die gemessene Auslastung reproduziert. Es ist zu beachten, dass die CPUDBSU-Konstante keinen Einfluss auf das IO-System der Datenbank besitzt. Für das IO-System müssen weitere Kalibrierungs-Maßnahmen getroffen werden. Hierzu existiert in den Settings einer Konfiguration der Parameter IODBBSU, der die Anzahl der zu verarbeitenden IOs pro DBSU vorgibt.

(c) Methodik zur Modellierung von Zentralserversystemen

R/3-Systeme mit nur einem physikalischen Server, der für Applikationen und Datenbank genutzt wird, bezeichnet man als Zentralserversystem. Mit Hilfe des WLPSizers können Zentralserversysteme modelliert werden. Bei Eingabe einer neuen Konfiguration wird bereits als Basiskonfiguration ein Zentralserversystem erstellt. Eine Besonderheit bei der Modellierung dieser Systeme ist, dass im Modell die CPU-Zeit-Verbräuche für Applikationen

und Datenbank zu einem Verbrauch (Bedienzeit) zusammengefasst werden, da die Aufträge nur von einem Server bearbeitet werden. Die Modell-Ergebnisse können somit nicht mehr nach Datenbank- und Applikations-Zeit unterschieden werden. Das heißt im Speziellen, dass die Spalte AppTi der Ergebnis-Sichten nicht belegt ist (= 0) und in der Spalte DBTi die Summe der Verweilzeiten am Application- und DB-Server angezeigt wird.

Um auch für Zentralserversysteme eine nach Applikation und Datenbank getrennte Ergebnis-Darstellung zu erhalten, können die Workload-Profile nach folgendem Schema modifiziert werden:

Jedes auf dem Zentralserver zu verarbeitende Workload-Profil WLP wird in die folgenden Profile aufgeteilt:

@ *WLP-DB:* Profil enthält die Spalten DBSU und DBKB

@ *WLP-CPU:* Profil enthält die Spalten CPUTI, OTHKB und CLBY

Es werden somit aus einem vom WLPMaker erstellten Workload-Profil WLP zwei neue Profile WLP-DB und WLP-CPU erstellt. Beide Profile besitzen jeweils die Spalten für Tasktyp, Komplexität und Ratio und auch die gleichen Durchsatzanforderungen. Die jeweils ausgelassenen Spalten werden mit Nullen aufgefüllt.. Anschließend werden die Profile in die Konfiguration eingefügt, wobei die zugehörigen Workloads WL-DB und WL-CPU die gleichen Eigenschaften besitzen müssen, das heißt gleicher Durchsatz und gleicher Workload-Typ.

Das Modell liefert anschließend Ergebnisse für beide Workloads WL-DB und WL-CPU. In der Server-View werden die jeweils zueinander korrespondierenden Ergebniszeilen der Workloads, also gleicher Tasktyp und Komplexitätsklasse, spaltenweise zusammengefasst. Es ergeben sich somit die folgenden neuen Ergebnisspalten:

@ *Thruput:* Die Durchsätze beider Workloads sind identisch. Für die Zusammenfassung der jeweils zugehörigen Zeilen werden die Duchsätze nicht summiert.

@ *RespTi*: Die Antwortzeiten der jeweiligen Zeilen werden addiert.

@ *AppTi:* Die AppTi entspricht der DBTi der WL-CPU.

@ *DBTi:* Die DBTi entspricht der DBTi der WL-DB.

@ *Stretch-Factor*: Die Stretch-Faktoren müssen neu berechnet werden.

Alle weiteren Spalten können additiv zusammengefasst werden. Die berechneten Ergebnis-Spalten entsprechen dann der gesuchten Ergebnis-Darstellung getrennt nach AppTi und DBTi. Der bei dieser Vorgehensweise existierende Nachteil ist die durch die Aufteilung der Workloads entstehende Auftragsverdopplung im Modell.

(d) Vorgehensweisen zur Prognosemodellierung

Die im WLPSizer generierten Modelle können zum einen ein bereits existierendes System in seiner Topologie und Performance (Antwortzeiten, Auslastungen und Durchsätze) nachbilden (Basismodellierung) oder aufbauend auf ein solches Modell Prognosen für zukünftige Szenarien bieten. Mit Hilfe des WLPSizers können Prognosen in den folgenden Aufgabengebieten erstellt werden:

- R/3-Release-Wechsel
- Hardware-Upgrade
- Zukünftige Laststeigerungen

Die Anwendungsmöglichkeiten des WLPSizers in diesen Gebieten werden in den folgenden Abschnitten beschrieben.

R/3-Release-Wechsel

Ein R/3-Release-Wechsel bedeutet häufig einen Ressourcenmehrbedarf an CPU-Zeit, DB-Zeit und IO. Entscheidend ist somit die Frage, ob das System nach vollzogenem Release-Wechsel die Dienstgüteanforderungen bez. Durchsatz und Antwortzeit weiterhin einhält. Mittels WLPSizer kann der bevorstehende Release-Wechsel auf zwei verschiedene Arten modelliert werden:

- *Anpassung der SAPS-Werte*:
  Die Server werden im Modell mit Leistungskennzahlen in SAPS beschrieben, die aufgrund von Benchmarkmessungen ermittelt wurden. Die SAPS-Definition ist abhängig vom zugrunde liegenden R/3-Release, so dass für jeden Server in Abhängigkeit des R/3-Release unterschiedliche SAPS-Zahlen vorliegen (siehe Tabelle 03-13).

HAL 2001							
	#CPU	1	2	3	4	5	6
R/3 Release	Faktor						
4.0B	35%	390	656	921	1187	1452	1718
4.5	15%	332	558	783	1009	1234	1460
4.6	25%	249	418	587	757	926	1095

Tabelle 03-13: SAPS-Tabelle für verschiedene R/3-Releases

Im Modell können somit die jeweiligen Server mit Hilfe der Bibliotheken ausgetauscht werden (siehe Abbildung 03-30). Es existieren bereits verschiedene CPU-Bibliotheken für die R/3-Releases.

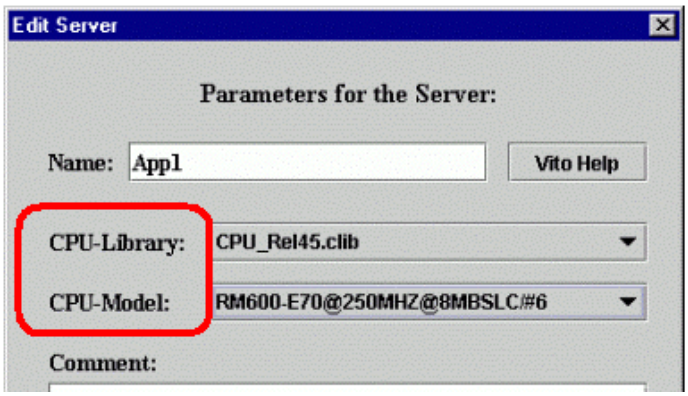


Abbildung 03-30: Verringerung der SAPS-Leistungen für den Release-Wechsel

@ *Erhöhung der CPU-Zeit-Mehrbedarfe für die Dialogschritte:*
Auf Basis von Benchmarkmessungen können für die Business Components eines R/3-Systems Faktoren berechnet werden, die den CPU-Zeit-Mehrbedarf für Dialog, Update und DB im Falle eines Release-Wechsels angeben (siehe Tabelle 03-14). Es kann somit im Vergleich zur vorherigen Methode eine detaillierte Parametrisierung des Modells erfolgen.

R/3-Modul	DIA	UPD	DB
FI	1,47	1,00	1,14
MM	1,13	1,03	1,03
PP	1,50	1,00	1,67
SD	1,48	1,01	1,20
WM	1,46	0,91	1,26

Tabelle 03-14: Faktoren für CPU-Zeiten der App.- und DB-Server (Rel. 4.5B nach 4.6B)

Die CPU-Zeit-Mehrbedarfe werden im Modell durch Modifikation der Workloads berücksichtigt, indem die Spalte CPUTI der Workloads mit dem entsprechenden Faktor multipliziert wird (siehe Abbildung 03-31).

T	C	ratio	CPUTI [ms]	DBSU	DBKB [KB]	RespTime [ms]	DBTime [ms]	LockTime [ms]	QueueTime [ms]
D	1	0,541	14,654	70,645	0,476	124,640	108,750	0,145	0,201
D	2	0,094	64,584	435,918	15,739	487,664	418,977	1,681	0,165
D	3	0,040	294,171	1.784,767	86,423	2.718,844	2.389,842	4,483	0,175
D	4	0,005	3.860,400	18.673,049	2.380,628	20.357,157	15.349,553	21,356	0,262
D	5	0,001	26.929,687	535.277,125	25.285,747	235.416,309	213.958,521	43,875	0,000
U	1	0,048	5,645	131,894	0,713	13,688	8,482	0,003	0,229
U	2	0,178	25,276	596,005	31,251	63,886	41,722	0,221	0,252
U	3	0,026	85,801	1.526,503	176,137	586,209	507,172	0,265	0,224
U	4	0,001	329,327	5.748,615	467,253	1.432,547	1.148,376	0,305	0,000
U	5	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
B	1	0,000	0,000	171,000	0,132	27,543	15,689	0,214	0,000
B	2	0,044	45,154	336,740	29,762	270,996	207,950	0,048	0,082
B	3	0,007	128,227	1.974,489	45,324	736,797	413,915	20,016	1,359
B	4	0,001	911,184	11.760,895	242,769	3.275,340	2.372,602	52,653	0,000
B	5	0,000	64.531,248	255.937,000	259.131,063	224.584,736	204.016,992	3,062	0,000
O	1	0,009	18,186	68,803	0,030	2.039,263	223,247	0,000	1.792,123
O	2	0,004	44,271	406,646	24,806	2.414,767	31,334	0,000	588,521
O	3	0,001	164,063	1.825,583	340,089	17.878,779	176,282	0,000	7.627,583
O	4	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
O	5	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
*	*	1,000	77,500	795,529	64,972	575,250	453,603	0,811	26,246

Abbildung 03-31: Erhöhung der CPU-Zeit-Mehrbedarfe für die Dialogschritte

Die CPU-Zeit-Mehrbedarfe für die Datenbank wurden durch Änderung der CPUDB-SU-Konstante realisiert. Zur Veranschaulichung der wachsenden Ressourcenansprüche bei steigendem R/3-Release-Wechsel, wurden in der folgenden Abbildung 03-32 die SAPS-Leistungen der Server den R/3-Release-Wechseln gegenübergestellt.

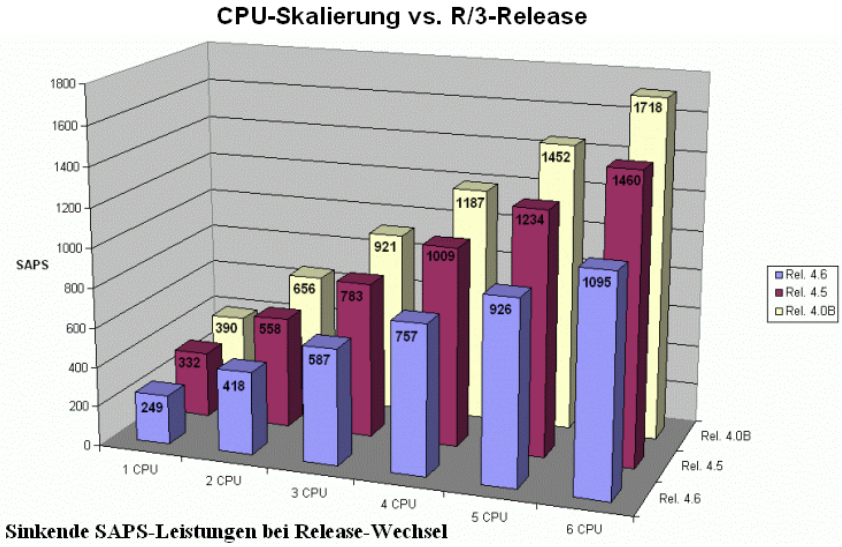


Abbildung 03-32: SAPS vs. R/3-Release

Hardware-Upgrade

Mit den Release-Wechseln eng gekoppelt sind die Hardware-Upgrades, die auch z.B. durch Inbetriebnahme weiterer Business-Components bzw. einen User-Anstieg verursacht werden können. Ein Hardware-Upgrade kann in zwei Fälle unterschieden werden:

@ *Vertikaler Upgrade:*
Die Modellkomponenten werden durch leistungsgsstärkere Bausteine ersetzt. So könnten zum Beispiel in einem Server weitere CPUs eingesetzt oder schnellere Festplatten im IO-System verwendet werden. Der Austausch der Modellkomponenten wird im WLPSizer mit Hilfe der Bibliotheken realisiert (siehe Abbildung 03-33).

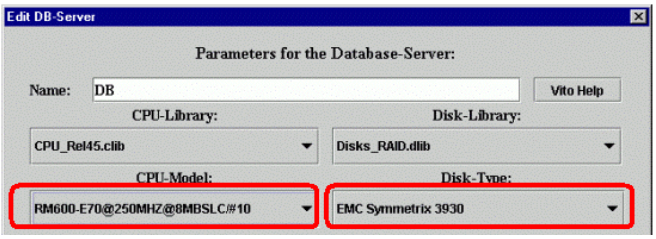


Abbildung 03-33: Vertikaler Upgrade am Beispiel des DB-Servers

@ *Horizontaler Upgrade:*
Das Modell wird durch weitere Komponenten ergänzt. Es werden zum Beispiel weitere Application-Server in das Modell eingefügt, die für eine verbesserte Lastverteilung genutzt werden können.

Zukünftige Laststeigerungen

Die Modellierung zukünftiger Laststeigerungen kann in drei Arbeitsschritte unterteilt werden:

@ *Lastprognose:*
Unter Lastprognose versteht man die Beschreibung der zukünftigen Last, wie zum Beispiel eine allgemeine Durchsatzsteigerung von 100 % (DS/h) oder die Beschreibung des Lastprofils einer zukünftigen Business-Component mit zu erwartenden Durchsatzanforderungen.

@ *Erstellung des Lastmodells:*
Das Lastmodell wird durch Modifikation bzw. Definition neuer Workload-Profile erstellt bzw. durch Veränderung der in den Workloads definierten Durchsätze. Der WLPSizer bietet hierzu die Funktion Experiment zur Erhöhung der Durchsätze aller Workloads mittels Eingabe eines Faktors.

@ *Modellexperimente:*
Modellexperimente dienen zum Beispiel der Untersuchung von einzuhaltenden Dienstgüten, wie zum Beispiel RespTime < 2 Sekunden für Dialogschritte der Komplexität 3. Weiterhin können notwendige Upgradevarianten betrachtet werden für zum Beispiel die Erhöhung der CPU-Anzahl oder die Hinzunahme weiterer Application-Server.

Die Ergebnisse der Modellexperimente können dann wie in Abbildung 03-34 dargestellt werden. Für einen Server werden unter verschiedenen Lastszenarien (Z-Achse) die Antwortzeiten für die Dialogschritte D2 und D3 (Y-Achse) für verschiedene Konfigurationsmöglichkeiten (X-Achse) dargestellt..

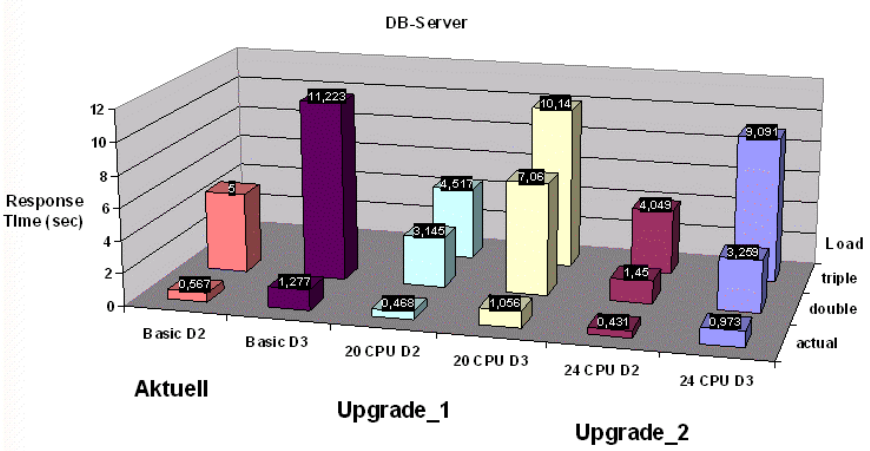


Abbildung 03-34: Prognoseergebnisse für D2 und D3

03.07 Beispiele

Die folgenden Beispiele beschreiben die Verwendung des WLPSizers insbesondere unter Anwendung der im vorherigen Abschnitt präsentierten Vorgehensweisen.

(a) Modellierung eines Zentralserversystems

Es wird ein Zentralserversystem bestehend aus einem Server RM600 E70 (250 MHZ) mit 4 Prozessoren und einer angeschlossenen Festplatte (10000 RPM, 80 MB/s) betrachtet. Der Server wird über ein Netz (Fast Ethernet 100 Mbit/s) mit einer Clientgroup verbunden. Ausgehend von der Clientgroup wird eine auf dem DB-Server zu verarbeitende Standard-Workload definiert. Das Workload-Profil wurde zur Vereinfachung selbst definiert und entspricht somit keiner realen Last. Es wurden nur Verbräuche für den Tasktyp Dialog definiert:

TC	ratio	CPUTI	DBSU	DBKB	OTHKB	CLBY
D1	0,50	50	60	0,5	0,1	1000
D2	0,35	100	200	10	0,5	1500
D3	0,15	250	1000	50	1,5	2000
…	…	…	…	…	…	…

Tabelle 03-15: Workload für das Zentralserver-Modell

Alle weiteren Einträge des Workload-Profils sind Null. Der für die Workload zugrunde liegende Durchsatz wurde auf 30 000 Dialogschritte pro Stunde festgelegt.

Es ergibt sich für dieses Modell eine Auslastung pro Prozessor von 56,06 % für den DB-Server und 31,26 % für das IO-System. Die folgende Abbildung 03-35 stellt die Modellergebnisse mit Hilfe der Server-View dar. Wie bereits beschrieben, werden die Antwortzeiten der Dialogschritte für ein Zentralserversystem nicht nach Applikations- und Datenbankzeit unterschieden, so dass beide Zeiten zusammengefasst in der Spalte DBTI dargestellt werden.

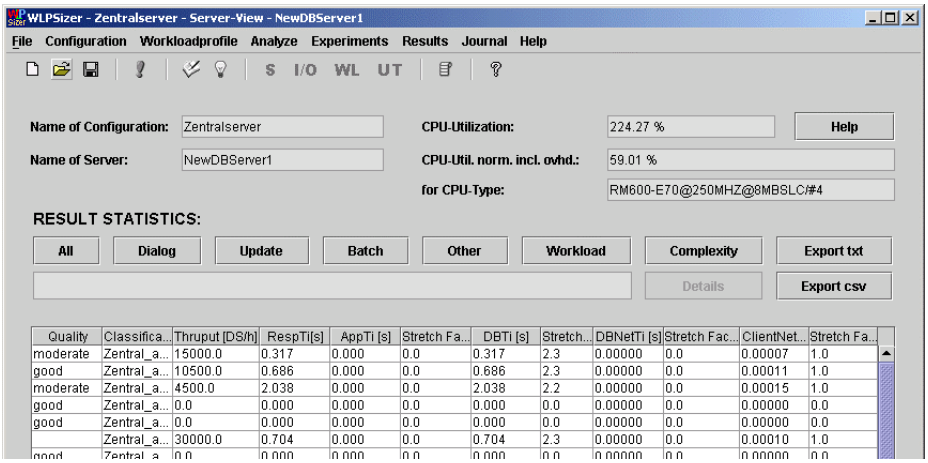

WLPSizer - Zentralserver - Server-View - NewDBServer1
File Configuration Workloadprofile Analyze Experiments Results Journal Help
S I/O WL UT
Name of Configuration: Zentralserver
Name of Server: NewDBServer1
CPU-Utilization: 224.27 %
CPU-Util. norm. incl. ovhd.: 59.01 %
for CPU-Type: RM600-E70@250MHZ@8MBSLC/#4
Help
RESULT STATISTICS:
All Dialog Update Batch Other Workload Complexity Export txt
Details Export csv
Quality Classifica... Thruput [DS/h] RespTi[s] AppTi [s] Stretch Fa... DBTi [s] Stretch... DBNetTi [s] Stretch Fac... ClientNet... Stretch Fa...
moderate Zentral_a... 15000.0 0.317 0.000 0.0 0.317 2.3 0.00000 0.0 0.00007 1.0
good Zentral_a... 10500.0 0.686 0.000 0.0 0.686 2.3 0.00000 0.0 0.00011 1.0
moderate Zentral_a... 4500.0 2.038 0.000 0.0 2.038 2.2 0.00000 0.0 0.00015 1.0
good Zentral_a... 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0
good Zentral_a... 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0
Zentral_a... 30000.0 0.704 0.000 0.0 0.704 2.3 0.00000 0.0 0.00010 1.0

Abbildung 03-35: Server-View für Zentralserver-Modell mit einer Workload

Zur Unterscheidung der Datenbank- und Applikationszeit wird die im vorigen Abschnitt 03.06 „Vorgehensweisen zur Prognosemodellierung“ beschriebene Methodik zur Modellierung von Zentralserversystemen angewandt. Das im Modell eingebundene Workload-Profil wird in zwei Profile WLP-DB und WLP-CPU aufgeteilt (siehe Tabelle 03-16 und Tabelle 03-17).

TC	ratio	CPUTI	DBSU	DBKB	OTHKB	CLBY
D1	0,50	0	60	0,5	0	0
D2	0,35	0	200	10	0	0
D3	0,15	0	1000	50	0	0
…	…	…	…	…	…	…

Tabelle 03-16: Workload-DB

TC	ratio	CPUTI	DBSU	DBKB	OTHKB	CLBY
D1	0,50	50	0	0	0,1	1000
D2	0,35	100	0	0	0,5	1500
D3	0,15	250	0	0	1,5	2000
…	…	…	…	…	…	…

Tabelle 03-17: Workload-CPU

Die in der Konfiguration vorhandene Workload wird ersetzt durch die Definition von zwei neuen Workloads, denen die Workload-Profile Workload-DB und Workload-CPU zugrunde liegen. Die bei Lösung des neuen Modells resultierenden Auslastungen sind bei 56,06 % für den DB-Server und 31,26 % für das IO-System verblieben. Die Antwortzeiten für die zwei neuen Workloads werden in der folgenden Abbildung 03-36 dargestellt.

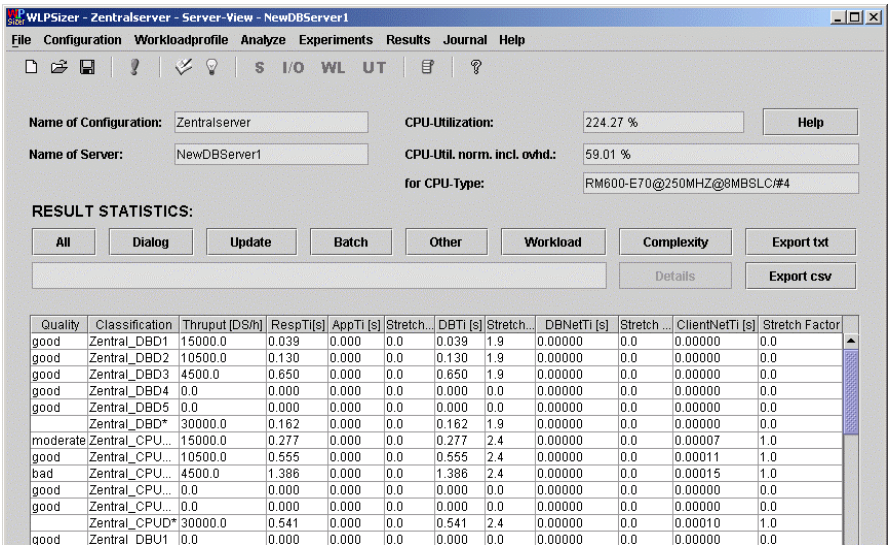

WLPSizer - Zentralserver - Server-View - NewDBServer1
File Configuration Workloadprofile Analyze Experiments Results Journal Help
S I/O WL UT
Name of Configuration: Zentralserver
Name of Server: NewDBServer1
CPU-Utilization: 224.27 %
CPU-Util. norm. incl. ovhd.: 59.01 %
for CPU-Type: RM600-E70@250MHZ@8MBSLC/#4
Help
RESULT STATISTICS:
All Dialog Update Batch Other Workload Complexity Export txt
Details Export csv
Quality Classification Thruput [DS/h] RespTi[s] AppTi [s] Stretch... DBTi [s] Stretch... DBNetTi [s] Stretch ... ClientNetTi [s] Stretch Factor
good Zentral_DBD1 15000.0 0.039 0.000 0.0 0.039 1.9 0.00000 0.0 0.00000 0.0
good Zentral_DBD2 10500.0 0.130 0.000 0.0 0.130 1.9 0.00000 0.0 0.00000 0.0
good Zentral_DBD3 4500.0 0.650 0.000 0.0 0.650 1.9 0.00000 0.0 0.00000 0.0
good Zentral_DBD4 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0
good Zentral_DBD5 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0
Zentral_DBD* 30000.0 0.162 0.000 0.0 0.162 1.9 0.00000 0.0 0.00000 0.0
moderate Zentral_CPU... 15000.0 0.277 0.000 0.0 0.277 2.4 0.00000 0.0 0.00007 1.0
good Zentral_CPU... 10500.0 0.555 0.000 0.0 0.555 2.4 0.00000 0.0 0.00011 1.0
bad Zentral_CPU... 4500.0 1.386 0.000 0.0 1.386 2.4 0.00000 0.0 0.00015 1.0
good Zentral_CPU... 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0
good Zentral_CPU... 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0
Zentral_CPUD* 30000.0 0.541 0.000 0.0 0.541 2.4 0.00000 0.0 0.00010 1.0
good Zentral_DBU1 0.0 0.000 0.000 0.0 0.000 0.0 0.00000 0.0 0.00000 0.0

Abbildung 03-36: Server-View des Zentralserver-Modells mit getrennten Workloads

Die für WLP-CPU angegebenen Zeiten der Spalte DBTI entsprechen den gesuchten Applikations-Zeiten APPTI. Die Ergebnisse können nun wieder zusammengefasst werden. Die

Durchsätze werden hierbei beibehalten, APPTI und DBTI wie beschrieben zugewiesen und die übrigen Zeilen addiert. Es ergeben sich somit die folgenden Ergebnisse:

TC	Thruput	RespTi [s]	AppTi [s]	DBTi [s]	App-DB-NetTi [s]	Client-NetTi [s]
D1	15000	0,316	0,277	0,039	0	0,00007
D2	10500	0,685	0,555	0,130	0	0,00011
D3	4500	2,036	1,386	0,650	1	0,00015
…	…	…	…	…	…	…

Tabelle 03-18: Zusammengefasste Ergebnisse des Modells mit getrennten Workloads

Die Berechnung der Stretch-Faktoren wurde ausgelassen. Die berechneten Werte aus Tabelle 03-18 können nun mit den Ergebnissen des ursprünglichen Modells in Abbildung 03-35 verglichen werden. Bis auf geringe Abweichungen sind die Ergebnisse für alle Spalten, mit Ausnahme von AppTi und DBTi, identisch.

(b) Modellierung eines 3-tier R/3-Systems

Das folgende Modell beschreibt eine R/3-Konfiguration bestehend aus zwei Applikations-Servern, einem Datenbank-Server (verbunden mit vier Festplatten) und drei Clientgroups. Der Datenbankserver ist über einen FDDI-Ring und die Clientgroups über ein Fast Ethernet (10 Mbit/s) mit den Applikations-Servern verbunden.

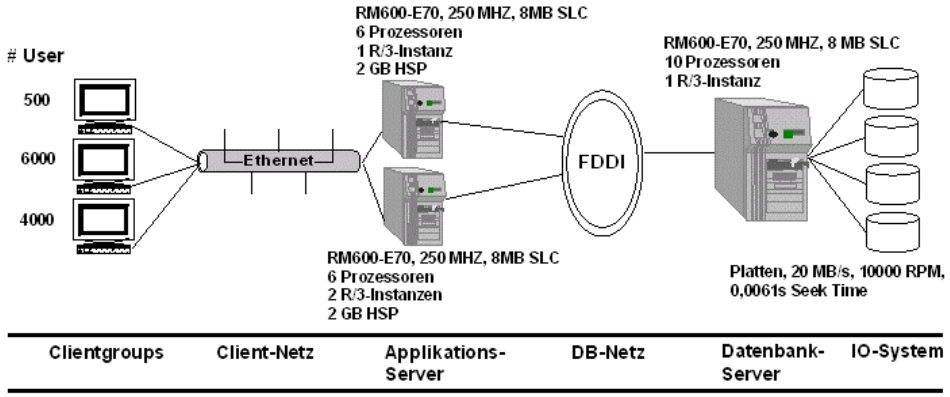
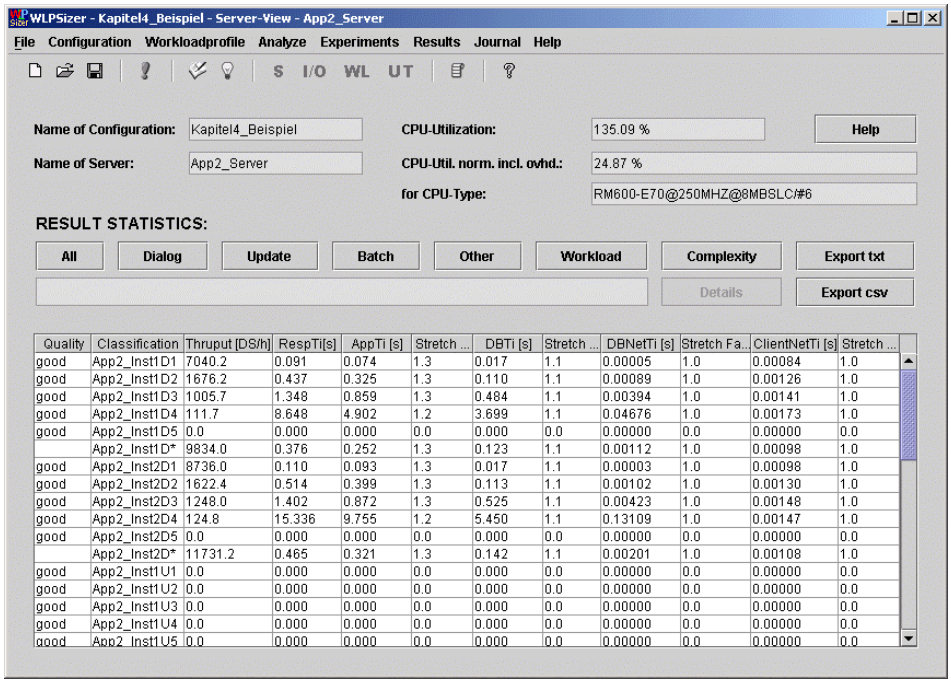


Abbildung 03-37: 3-tier R/3-System

Bei Lösung des Modells ergeben sich Auslastungen für die Stationen des Modells von 0,88 % für die Platten, 12,86 % für den DB-Server, 11,42 % für den Applikations-Server 1 und 22,51 % für den Applikations-Server 2. Die Auslastungen der Netze werden für dieses Beispiel nicht betrachtet. Aufgrund der geringen Auslastungen besitzen die Dialogschritte aller Tasktypen Antwortzeiten der Dienstgüte good. So liegen zum Beispiel die Antwortzeiten für Dialogschritte der Komplexitätsklasse drei unterhalb von 1,5 Sekunden (siehe Abbildung 03-38).


WLPSizer - Kapitel4_Beispiel - Server-View - App2_Server
File Configuration Workloadprofile Analyze Experiments Results Journal Help
S I/O WL UT
Name of Configuration: Kapitel4_Beispiel
Name of Server: App2_Server
CPU-Utilization: 135.09 %
CPU-Util. norm. incl. ovhd.: 24.87 %
for CPU-Type: RM600-E70@250MHZ@8MBSLC/#6
Help
RESULT STATISTICS:
All Dialog Update Batch Other Workload Complexity Export txt
Details Export csv

Quality	Classification	Thruput [DS/h]	RespTi[s]	AppTi [s]	Stretch ...	DBTi [s]	Stretch ...	DBNetTi [s]	Stretch Fa...	ClientNetTi [s]	Stretch ...
good	App2_Inst1D1	7040.2	0.091	0.074	1.3	0.017	1.1	0.00005	1.0	0.00084	1.0
good	App2_Inst1D2	1676.2	0.437	0.325	1.3	0.110	1.1	0.00089	1.0	0.00126	1.0
good	App2_Inst1D3	1005.7	1.348	0.859	1.3	0.484	1.1	0.00394	1.0	0.00141	1.0
good	App2_Inst1D4	111.7	8.648	4.902	1.2	3.699	1.1	0.04676	1.0	0.00173	1.0
good	App2_Inst1D5	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
	App2_Inst1D*	9834.0	0.376	0.252	1.3	0.123	1.1	0.00112	1.0	0.00098	1.0
good	App2_Inst2D1	8736.0	0.110	0.093	1.3	0.017	1.1	0.00003	1.0	0.00098	1.0
good	App2_Inst2D2	1622.4	0.514	0.399	1.3	0.113	1.1	0.00102	1.0	0.00130	1.0
good	App2_Inst2D3	1248.0	1.402	0.872	1.3	0.525	1.1	0.00423	1.0	0.00148	1.0
good	App2_Inst2D4	124.8	15.336	9.755	1.2	5.450	1.1	0.13109	1.0	0.00147	1.0
good	App2_Inst2D5	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
	App2_Inst2D*	11731.2	0.465	0.321	1.3	0.142	1.1	0.00201	1.0	0.00108	1.0
good	App2_Inst1U1	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Inst1U2	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Inst1U3	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Inst1U4	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Inst1U5	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0

Abbildung 03-38: Ergebnis-Sicht (Server-View von App2)

Hintergrundlasten

Das Modell wird nun um weitere Lasten erweitert. Ziel ist es hierbei, den Datenbank-Server und das IO-System mit weiterer NON-SAP-Last zu belasten. Die Erweiterung des Modells geschieht mit Hilfe der Hintergrundlast, die im WLPSizer als Workload-Typ ausgewählt werden kann. Es muss hierzu die entsprechende Last als Workload-Profil beschrieben werden. Für die CPU des Datenbank-Servers ergibt sich folgendes Profil:

TC	ratio	CPUTI	DBSU	DBKB	OTHKB	CLBY
D1	1,0	400	0	0	0	0
…	…	…	…	…	…	…

Tabelle 03-19: Hintergrundlast für die Datenbank-Server-CPUs

Die in Tabelle 03-19 dargestellte Lastbeschreibung für die NON-SAP-Hintergrundlast kann mit Hilfe einer Klasse von Dialogschritten (entspricht einer Zeile des Profils) beschrieben werden. Alle weiteren Zeilen sind Null. In Abhängigkeit der zu belastenden Station müssen nur die Spalten belegt werden, die auch zur Auslastung der Station beitragen. In diesem Fall sind die CPUs des Datenbank-Servers das Ziel der Hintergrundlast. Für CPUs ist vorwiegend die Spalte CPUTI relevant. Für den Datenbank-Server könnte auch die Spalte DBSU zur CPU-Belastung genutzt werden, da mittels der CPUDBSU-Konstante die aus den DBSU resultierende CPU-Belastung errechnet wird. Für das IO-System wird die folgenden Lastbeschreibung verwendet:

TC	ratio	CPUTI	DBSU	DBKB	OTHKB	CLBY
…	…	…	…	…	…	…
D3	1,0	0	1500	0	0	0
…	…	…	…	…	…	…

Tabelle 03-20: Hintergrundlast für das IO-System des Datenbank-Servers

Wie bereits beschrieben, wird auch hier die Last für das IO-System mittels einer Klasse von Dialogschritten beschrieben. Alle weiteren Einträge sind Null. Auf Basis der DBSU werden dann mittels der Konstante IODBSU (Settings-Parameter) die zu verarbeitenden IOs berechnet.

Die Eingabe der Hintergrundlast erfolgt über den Workload-Dialog. In diesem wird nach Einlesen des Profils der Workload-Typ Background ausgewählt. Anschließend wird für die Datenbank-CPUs der Datenbank-Server und für das IO-System die jeweilige Festplatte als Ziel festgelegt (siehe Abbildung 03-39). Für das IO-System wird somit für alle vier Festplatten die gleiche Workload als Hintergrundlast in das Modell eingefügt.

Über die Schaltfläche Additional Parameters werden die Anzahl der User, die die Hintergrundlast produzieren, und die Durchsatzsteuerung angegeben. Es kann hierbei eingestellt werden, dass die für die Workloads angegebenen Durchsätze erzwungen werden oder dass

eingegebene Denkzeiten zwischen den Dialogschritten eingehalten werden sollen. Es werden für dieses Modell 1000 User festgelegt und die Durchsatzsteuerung aktiviert.

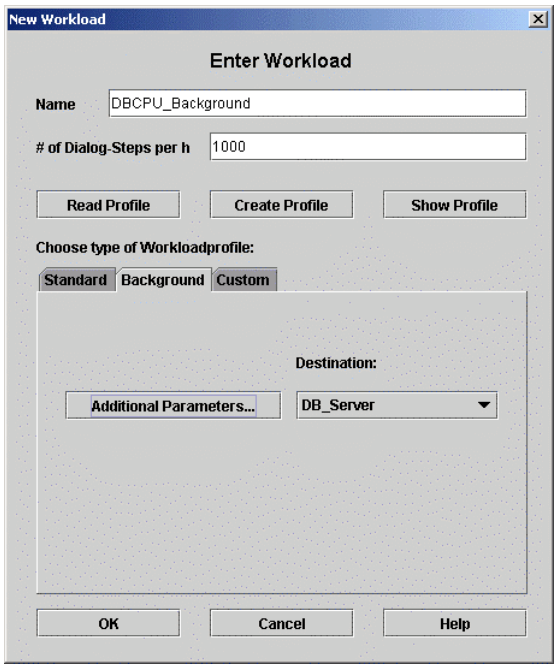


Abbildung 03-39: Hintergrundlast-Dialog für die DB-Server-CPUs

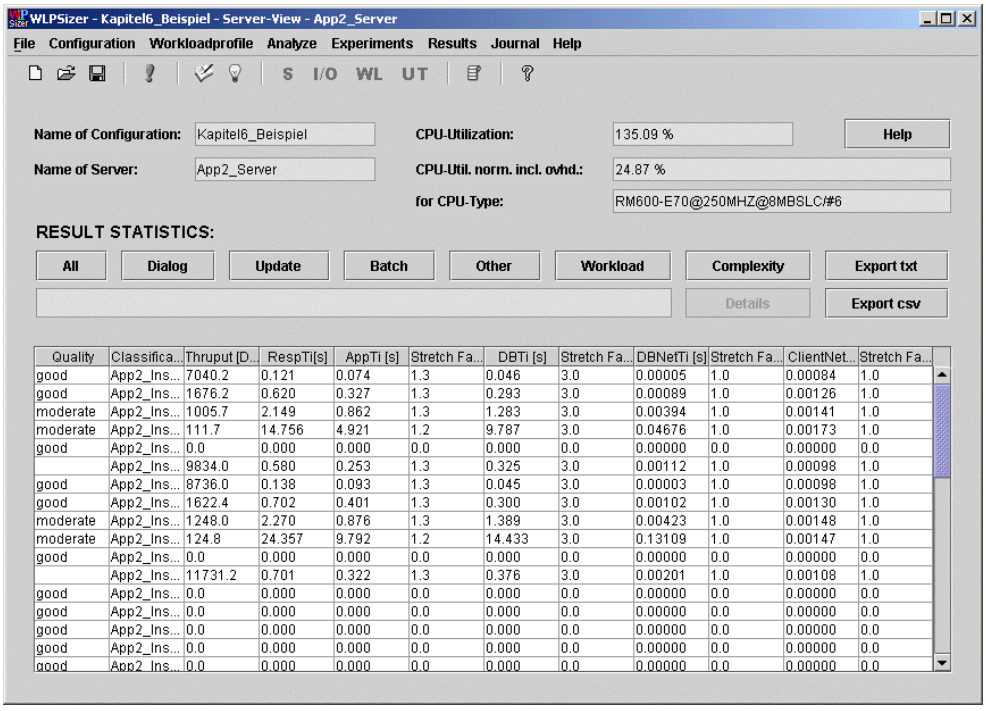
Abschließend muss für jede Hintergrundlast ein Durchsatz eingestellt werden, der die gewünschte Auslastung für die Datenbank-CPUs und das IO-System erzielt. Im vorliegenden Fall wurden die folgenden Durchsätze gewählt und die nebenstehenden Auslastungen gewonnen:

Station	**Dialogschritte pro Stunde**	**Auslastung**
DB-CPUs	15000	57,12 %[31]
Festplatten 1,2	40000	3,79 %
Festplatten 3,4	60000	5,25 %

Tabelle 03-21: Durchsätze für die Hintergrundlasten

[31] Auslastung pro Prozessor

Die Antwortzeiten der Dialogschritte (Tasktyp Dialog) stiegen durch die erhöhte Auslastung an und verloren zum Teil ihre Dienstgüte good, so dass zum Beispiel die Dialogschritte der Komplexitätsklasse drei von einer durchschnittlichen Antwortzeit unter 1,5 Sekunden auf über 2 Sekunden anstiegen und somit Dienstgüte moderate erhielten (siehe Abbildung 03-40).



Quality	Classifica...	Thruput [D...	RespTi[s]	AppTi [s]	Stretch Fa...	DBTi [s]	Stretch Fa...	DBNetTi [s]	Stretch Fa...	ClientNet...	Stretch Fa...
good	App2_Ins...	7040.2	0.121	0.074	1.3	0.046	3.0	0.00005	1.0	0.00084	1.0
good	App2_Ins...	1676.2	0.620	0.327	1.3	0.293	3.0	0.00089	1.0	0.00126	1.0
moderate	App2_Ins...	1005.7	2.149	0.862	1.3	1.283	3.0	0.00394	1.0	0.00141	1.0
moderate	App2_Ins...	111.7	14.756	4.921	1.2	9.787	3.0	0.04676	1.0	0.00173	1.0
good	App2_Ins...	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
	App2_Ins...	9834.0	0.580	0.253	1.3	0.325	3.0	0.00112	1.0	0.00098	1.0
good	App2_Ins...	8736.0	0.138	0.093	1.3	0.045	3.0	0.00003	1.0	0.00098	1.0
good	App2_Ins...	1622.4	0.702	0.401	1.3	0.300	3.0	0.00102	1.0	0.00130	1.0
moderate	App2_Ins...	1248.0	2.270	0.876	1.3	1.389	3.0	0.00423	1.0	0.00148	1.0
moderate	App2_Ins...	124.8	24.357	9.792	1.2	14.433	3.0	0.13109	1.0	0.00147	1.0
good	App2_Ins...	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
	App2_Ins...	11731.2	0.701	0.322	1.3	0.376	3.0	0.00201	1.0	0.00108	1.0
good	App2_Ins...	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Ins...	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Ins...	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Ins...	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0
good	App2_Ins...	0.0	0.000	0.000	0.0	0.000	0.0	0.00000	0.0	0.00000	0.0

Abbildung 03-40: Server-View für das Modell mit Hintergrundlast

03.08 Literatur

[1] Schneider, Thomas; *„SAP R/3-Performance-Optimierung“*; Addison Wesley, 1999

[2] Universität Essen; *„WLPSizer 1.5 – Werkzeug zur Kapazitätsplanung von SAP R/3-Systemen“*; 2001

[3] University of Essen; *“Capacity Planning and Performance Evaluation with the Tool VITO”*; December 2000

[4] Buck-Emden, Rüdiger; *„Die Technologie des SAP R/3-Systems“*; Addison Wesley, 1999

[5] Gustafson, J.L.; *“Reevaluating Amdahl´s Law”*; www.scl.ameslab.gov/Publications/AmdahlsLaw/Amdahls.html; 1988

[6] Siemens ATD IT PS Mhm; *“myAMC.LNI User Guide”*, 2000

[7] Müller-Clostermann, B.; Totzauer, G.; *„TOTO Benutzerhandbuch, Version 1.0“*; 1997

Teil VII: Praxisberichte

Inhalt von Teil VII

KAPITEL 01 KAPAZITÄTSPLANUNG VON IT-SYSTEMEN 4

01.01 ÜBER KAPAZITÄTSMANAGEMENT UND KOSTEN 4
01.02 METHODEN ZUR KAPAZITÄTSPLANUNG 5
01.03 ANWENDER FORMULIEREN IHRE FORDERUNGEN 8

KAPITEL 02 DAS MAPKIT-VORGEHENSMODELL: KAPAZITÄTSPLANUNG EINES HETEROGENEN CLIENT/SERVER-SYSTEMS 9

02.01 EINFÜHRUNG UND MOTIVATION 9
02.02 VORGEHENSMODELLE ZUR KAPAZITÄTSPLANUNG 9
02.03 DAS MAPKIT VORGEHENSMODELL 10
02.04 FALLSTUDIE ZUM MAPKIT VORGEHENSMODELL 13
02.05 PLANUNG ZUKÜNFTIGER SZENARIEN 23
02.06 PERFORMANCE-PROGNOSEN UND DIMENSIONIERUNG 25
02.07 ZUSAMMENFASSUNG UND AUSBLICK 29
02.08 REFERENZEN 29

KAPITEL 03 E-MANAGEMENTLEITSTAND "LIGHT" 30

03.01 MOTIVATION UND EINFÜHRUNG 30
03.02 AUFGABENSTELLUNG 30
03.03 TECHNISCHES KONZEPT 31
03.04 TECHNISCHE REALISIERUNG 32
03.05 INSTALLATION UND BETRIEB 34
03.06 AUSBLICK 35
03.07 FAZIT 36
03.08 DANKSAGUNGEN 37

KAPITEL 04 MIT LAST- UND PERFORMANCE-TESTS ZUM EINSATZREIFEN SYSTEM 38

04.01 DIE AUFGABE 38
04.02 HOCHGESTECKTE ZIELE 39
04.03 DESIGN UND IMPLEMENTIERUNG VON SZENARIEN 39
04.04 STRUKTURIERUNG UND VERFÜGBARKEIT 40
04.05 DURCH REALITÄTSNAHE SIMULATION ZUR PRAXISREIFE 40
04.06 PILOTPHASE UND ROLL-OUT 43

KAPITEL 05 LAST- UND PERFORMANCE-TESTS AM BEISPIEL IT2000 44

05.01 EINFÜHRUNG UND MOTIVATION 44
05.02 AUTOMATISCHE LASTGENERIERUNG 45
05.03 DURCHFÜHRUNG VON AUTOMATISIERTEN LAST- UND PERFORMANCE-TESTS 48
05.04 LAST- UND PERFORMANCETEST ATLAS 50

KAPITEL 01 KAPAZITÄTSPLANUNG VON IT-SYSTEMEN

SIEGFRIED GLOBISCH, KLAUS HIRSCH[1]

01.01 Über Kapazitätsmanagement und Kosten

Wohl dem IT-Administrator, der eine Windows-Installation zu verantworten hat. Bei allen Problemen, die das Microsoft-Betriebssystem ansonsten mit sich bringt – im Bereich der Kapazitätsplanung und des Performance Managements kann es sich der IT-Verantwortliche im Vergleich zu seinen Kollegen aus der Unix- und Mainframe-Welt recht leicht machen. Klemmt die Leistung an irgendeiner Stelle, wird einfach Hardware hinzugekauft. "Nachschiebe-Mentalität" nennen Experten wie Klaus Hirsch diese Vorgehensweise.

(a) Hardwareaufrüstung und Kosten

Der Kopf der Expertengruppe IT-Resource-Management innerhalb des Bereichs System-Integration bei Siemens Business Services (SBS) leitet ein Team, das im Kundenauftrag Schwachstellen und Engpässe in komplexen IT-Systemen aufspürt und Prognosen über den künftigen Ressourcenbedarf erstellt. NT-Kunden zählen selten zu seiner Klientel, eben wegen der besagten Nachschiebementalität, denn die Beratungsleistung der SBS-Truppe kann durchaus einige zehntausend Mark kosten. Für NT-Maschinen rechnet sich ein derartiger Service häufig nicht, es sein denn, sie beherbergen unternehmenskritische Applikationen. In der Regel laufen für das Kerngeschäft wichtige Applikationen jedoch in einer Unix- und Großrechnerumgebung, und diese Geräte wollen mit steigender Anwenderzahl und zunehmend intensiveren Nutzung regelmäßig aufgerüstet werden. Dabei entscheiden die IT-Verantwortlichen über Beträge, die im Mainframe-Umfeld die Millionen-Mark-Grenze übersteigen. "Bei einer Hardwareaufrüstung reden wir über Summen im siebenstelligen Bereich", vermittelt Siegfried Globisch von R+V Versicherung einen Eindruck von den Investitionsvolumina.

(b) Softwarekosten

"Das dicke Ende kommt jedoch mit den Softwarekosten, die sich auf das Zehnfache der Ausgaben für die Hardware belaufen können", weiß der Messexperte, der bei dem Versi-

[1] Der Beitrag ist in ähnlicher Form erschienen in Computerwoche Nr. 34/2000

cherungskonzern das Kapazitäts- und Performance-Management für die hausinternen Anwendungen betreut. Denn unabhängig davon, ob die auf dem Großrechner eingesetzten Tools von der Aufrüstung betroffen sind oder nicht: Der Lieferant hält aufgrund der CPU-abhängigen Lizenzierung die Hand auf. Die Änderungen in der IT-Installation wollen demnach gut überlegt sein, doch auf Hilfestellung seitens der Hardwarehersteller können sich die Anwender nicht verlassen.

(c) Mess- und Monitoring-Tools

Übereinstimmend stellen Globisch und Hirsch den für die Client/Server-Plattformen mitgelieferten Standardmess-Tools ein Armutszeugnis aus. Zwar sind diese Verfahren für ein Online-Monitoring ausgelegt, sie liefern aber häufig nicht die präzisen Aussagen, die man sich wünscht. Gerade aus Applikationssicht zeigen die Werkzeuge unübersehbare Schwächen. "Die Messsysteme stellen vielfach nur eine geringe Analogie zu den abgewickelten Geschäftsprozessen her", bemängelt SBS-Manager Klaus Hirsch. Dieser Zustand ist für den Spezialisten der R+V Globisch besonders ärgerlich, da bei der Versicherung das Kerngeschäft fast ausschließlich IT-basierend betrieben wird. Der Blick in die Applikationslogik bleibt den für die Qualitätssicherung verantwortlichen IT-Experten hingegen verwehrt.

01.02 Methoden zur Kapazitätsplanung

Für die Kapazitätsplanung müssen die IT-Administratoren daher auf andere Verfahren ausweichen. "Über den dicken Daumen", so Hirsch und Globisch unisono, ist die einfachste, billigste, aber auch ungenaueste Methode der Performance-Prognose. Sie kommt häufig bei IT-Systemen mit unkritischen Anwendungen zum Einsatz. Wenn dort etwa das Antwortzeitverhalten nicht mehr stimmt, werden zusätzliche Prozessoren und Hauptspeicher eingeschoben oder die Plattenkapazität erhöht, in der Hoffnung, die sich abzeichnenden Probleme damit zu beseitigen. Eine weitere relativ kostengünstige Möglichkeit ist der Rückgriff auf vorhandenen Erfahrungsschatz. Will der Anwender beispielsweise eine neue Anwendung einführen und hat eine ähnliche Applikation bereits im Haus, lässt sich aufgrund der Analogie zwischen neuer und vorhandener Anwendung der Ressourcenbedarf zumindest annähernd abschätzen.

Beide Methoden sind für unternehmenskritische Systeme mit hohem Stellenwert nicht empfehlenswert. Hier sind ausgefeiltere Ansätze gefragt, die aber allesamt eine gute Kenntnis der installierten DV sowie des aktuellen und erwarteten Nutzungsverhaltens voraussetzen. Gemeinsames Ziel der anspruchsvolleren Prognoseverfahren ist es, Engpässe und Überkapazitäten bei den zu tätigenden Hardwareinvestitionen gleichermaßen zu vermeiden. Letzteres kann sehr teuer werden und rechtfertigt unterm Strich die Aufwände für eine genauere Vorgehensweise.

Eine weit verbreitete, sehr effiziente Methode der vorausschauenden Planung ist die der analytischen Modellierung. Die dabei verwendeten Berechnungsverfahren müssen der Anforderung genügen, bei Änderungen der Last und Hardwarestruktur zuverlässige Aussagen zum veränderten Systemverhalten zu liefern.

(a) Analytische Modellierung für die vorausschauende Planung

Um die Genauigkeit eines Modells zu kontrollieren, wird die Ist-Situation anhand von Messwerten erfasst und mit den Ergebnisdaten des analytischen Modells verglichen. Wenn die Auslastungswerte und das Antwortzeitverhalten des reellen und des nachgebildeten Systems übereinstimmen, ist der nächste Schritt fällig. In Kenntnis der künftigen Nutzung lassen sich im Modell dann die Lastparameter variieren.

Wenn die Fachabteilung beispielsweise weiß, dass künftig auf Applikation A 400 und auf die Applikation B 50 Anwender zugreifen werden, das Verhältnis bislang jedoch 200 zu 100 betrug, können solche Angaben bei der analytischen Modellierung berücksichtigt werden. "Es lassen sich Lastparameter verändern und Hardwarebausteine ersetzen, um die Leistungsfähigkeit zu modifizieren. So ergeben sich Aussagen darüber, wie die IT-Installation in einem halben oder einem Jahr auszusehen hat, damit sie mit dem prognostizierten Nutzungsaufkommen zurechtkommt", erläutert Hirsch.

Unter der Leitung von Hirsch wird im Hause Siemens ein entsprechendes Produkt entwickelt und auch verkauft. Als Kunden kommen in erster Linie Firmen in Frage, für die es sich aufgrund der Größe der eigenen IT-Landschaft lohnt, "das Kapazitätsmanagement von spezialisierten Mitarbeitern aus den eigenen Reihen vornehmen zu lassen", so der SBS-Manager. Meistens erledigen Hirsch und seine Mitarbeiter derartige Bedarfsprognosen als Dienstleister für solche Unternehmen, denn der Umgang mit analytischen Modellierungsverfahren erfordert profundes Know-how, insbesondere aber auch die Kenntnis darüber, wann es notwendig wird, beispielsweise einen User-Benchmark den genannten Ansätzen vorzuziehen.

(b) User-Benchmarking

Bei der R+V Versicherung gibt es zwar einen Expertenstamm für die Kapazitätsplanung. Analytische Modelle kommen aber noch wenig zum Einsatz. Dort verwendet man für Last- und Performance-Tests von Anwendungen zunächst User-Benchmarks, ein Verfahren, das sich als eine synthetische Nachbildung eines definierten Lastaufkommens auf der Zielhardware, also einem realen System, beschreiben lässt. Um aussagekräftige Daten zu erhalten, sollte das Testsystem sämtliche repräsentative Datenbestände und vergleichbare Applikationen in einem funktionierenden Wirkungsgefüge umfassen.

Ein User-Benchmark ist zumeist kein Tagesgeschäft und kann sich zudem in Verbindung mit darauf aufbauenden Performance-Modellierungen laut R+V Expertem Globisch über einige Monate hinziehen. Die Dauer des Testszenarios und die Verwendung eines tatsächlichen Abbilds des Produktivsystems macht diese Form der Kapazitätsplanung zu einer teuren Variante, die aber laut Hirsch in der Regel sehr gute Ergebnisse liefert.

Doch bevor es so weit ist, müssen weitere Vorbereitungen getroffen werden, die sich in der Hirsch-Terminologie unter der Bezeichnung "Drehbuch" subsumieren lassen. Konkret heißt dies, dass Scripts erstellt werden, die die Tätigkeiten und Zugriffe der Benutzer auf das Produktivsystem nachbilden. "Analog dazu könnte man natürlich auch 1000 Benutzer mit Stoppuhr und persönlichem Ablaufplan vor ihre Terminals setzen, die nach einem definierten Muster Eingaben und Anfragen durchspielen", veranschaulicht Hirsch die Ablauflogik eines solchen Projekts. Das Drehbuch entsteht in Zusammenarbeit mit den Fachabteilungen, denn die müssen die relevanten Geschäftsprozesse und die zuordenbaren Applikationen identifizieren und typische Anwendungsszenarien entwickeln. Sinn und Zweck des so erstellten Testablaufs ist aber nicht allein die möglichst genaue Abbildung der realen IT-Nutzung, sondern auch die Reproduzierbarkeit einzelner Lastsituationen unter veränderten Rahmenbedingungen.

Denn die im User-Benchmark ermittelten Kenngrößen wie Antwortzeitverhalten, Ein- und Ausgabeverkehr sowie CPU-Belastung müssen sich einer definierten, nachgestellten Lastsituation zuordnen lassen. Ein wesentlicher Vorteil dieses Verfahrens ist zudem, dass es im Vergleich zum analytischen Modell Aussagen über ein mögliches, ansonsten nicht vorhersehbares Sperrverhalten von Applikationen und systemnaher Software liefern kann.

(c) Simulation

Eine Alternative, die man von Aussagekraft und Aufwand her zwischen analytischem Modell und User-Benchmark einordnen kann, sind simulative Verfahren. Hier werden die in einem realen IT-System ablaufenden Vorgänge nicht mehr durch Algorithmen, sondern durch analoge Modellereignisse nachvollzogen. Mit Simulationsmethoden können insbesondere statistische Werte in Bezug auf die untersuchten Leistungsgrößen gewonnen werden. Ein Beispiel hierfür sind so genannte Antwortzeitquantile. Der Nachteil: Während die Berechnung eines analytischen Modells wenige Sekunden benötigt, kann der gleiche Rechner bei der Simulation eines vergleichbaren Szenarios durchaus einige Tage mit seiner Rechenaufgabe beschäftigt sein. Bei Hirsch und seinen Mitarbeitern kommt dieses Verfahren nicht zum Einsatz, und auch Globisch von der R+V Versicherung lehnt es ab, aus einem einfachen Grund: "Viel zu teuer", urteilt er, "das können sich nur Hersteller leisten." Hirsch schätzt, dass derartige Simulationen bis zu 50mal teurer sein können als Berechnungen mit dem analytischen Modell.

(d) In der Praxis eingesetzte Verfahren

Aus diesem Grund, aber auch weil die SBS-Experten sehr viel Erfahrung mit diesen Verfahren haben, nutzen Hirsch und seine Mitarbeiter in einer Vielzahl von Projekten zur Performance-Prognose die Vorgehensweise anhand des analytischen Modells, wie auch den User-Benchmark. Die Kapazitätsmanagement-Mannschaft der R+V Versicherung verwendet hingegen zunächst einfache Formen von User-Benchmarks, wenn sie derartige Vorhaben stemmen wollen. In jedem Fall sind aber Produkte erforderlich, die die benötigten Messwerte in einer Mainframe- und Client/Server-Umgebung erheben. Zufrieden ist Globisch mit den Angeboten der Hersteller aber nicht. Im Client/Server-Umfeld, so seine Kritik, liefern derartige Tools nur die Basisdaten des Systems, zwar besser aufbereitet, aber nicht detaillierter. Benutzerbezogene Messwerte fehlen zumeist völlig. Mess-Tools für den Großrechner und den Unix-Bereich tragen häufig die gleiche Bezeichnung, kommunizieren untereinander aber nicht.

01.03 Anwender formulieren ihre Forderungen

Diese generell unbefriedigende Situation hat den SBS-Experten Hirsch veranlasst, zusammen mit einer Reihe von gleichgesinnten Partnern aktiv zu werden. Neben Kollegen aus diversen Siemens-Bereichen sind das Institut für Informatik an der Universität Essen sowie als mittelständisches Unternehmen Materna in Dortmund beteiligt. Im Rahmen eines beim Bundesministerium für Bildung und Forschung (BMBF) und der DLR in Berlin verankerten Förderprojektes wird die Thematik des Kapazitätsmanagements in heterogenen, verteilten IT-Systemen unter die Lupe genommen und nach methodischen Lösungsansätzen gesucht, die für den IT-Anwender und Betreiber in diesem Umfeld nützlich und hilfreich sind. Auch Globisch hat gehandelt und diese Themen unter dem Dach der Central Europe Computer Measurement Group (CE-CMG) in den Arbeitsgruppen "Messgrößen im Client/Server-Umfeld" sowie "Kapazitäts- und Performance-Management" aufgegriffen. Auf gemeinsam mit der Arbeitgruppe "Planung und Modellierung" durchgeführten Workshops werden unter anderem Forderungen an die Hersteller formuliert. Ob die Vorschläge aufgegriffen werden, ist ungewiss. Zumindest möchte der R+V Performance-Spezialist versuchen, dieses Thema auch auf europäischer Ebene der CMG zur Sprache zur bringen, um sich so Gehör bei den Anbietern zu verschaffen.

KAPITEL 02 DAS MAPKIT-VORGEHENSMODELL: KAPAZITÄTSPLANUNG EINES HETEROGENEN CLIENT/SERVER-SYSTEMS

CORINNA FLUES, JÖRG HINTELMANN

02.01 Einführung und Motivation

Trotz der zunehmenden Abhängigkeit des Unternehmenserfolgs von der Leistungsfähigkeit und der Verlässlichkeit der IT-Strukturen ist ein systematisches Vorgehen im Bereich Kapazitätsmanagement in den meisten Unternehmen völlig unzureichend etabliert. Zu den Aktivitäten des Kapazitätsmanagements gehören solche Maßnahmen, die darauf abzielen, die verfügbaren Systemressourcen so einzusetzen, dass eine bestmögliche Performance des Systems erreicht wird. Diese Aktivitäten werden auch als Performance-Tuning bezeichnet. Weiterhin umfasst es die Aktivitäten, die zu ergreifen sind, um IT-Systeme entsprechend zukünftiger Anforderungen zu dimensionieren, wobei die jeweils bestmögliche IT-Architektur von angemessener Größenordnung zu bestimmen ist. Dabei ist insbesondere die Einhaltung vertraglicher Vereinbarungen in Form von *Service Level Agreements* über Dienstgütemerkmale wie Transaktionsdurchsatz oder Antwortzeiten anzustreben.

In dem vorliegenden Artikel wird eine durchgängige Vorgehensweise zur Durchführung von Kapazitätsmanagement in heterogenen Client/Server-Systemen vorgestellt und exemplarisch an einer Fallstudie demonstriert.

02.02 Vorgehensmodelle zur Kapazitätsplanung

Vorgehensmodelle zum Kapazitätsmanagement in heterogenen verteilten Systemen sind nur selten in der einschlägigen Literatur zu finden. Berücksichtigt man den gesamten Prozess von der Sensibilisierung der Verantwortlichen in den Unternehmen bis hin zur Planung und Prognose der Systeme hinsichtlich ausgehandelter Dienstgüteparameter, so findet sich kein durchgängiges und umfassendes Vorgehensmodell. Die Vorgehensmodelle nach [Mena94] und [Jain91] beinhalten Strategien, die geeignet sind, Performance-Prognosen für Systeme zu erstellen, die Lastevolutionen ausgesetzt sind. Sie beschreiben die Aktivitäten Lasterfassung und Lastvoraussage, die Bildung von Systemmodellen sowie Methoden zur Performance-Prognose. Während Menascé eine explizite Validierungsphase zumindest für das Systemmodell vorsieht, fehlt dieser Aspekt bei Jain völlig. Beide Vorgehensmodelle

berücksichtigen lediglich technische bzw. mathematische Aspekte, die bei stochastischer Performance-Modellierung auftreten.

Zwei weitere Vorgehensmodelle, die den Schwerpunkt auf eine werkzeugbasierte Strategie legen, betrachten Kapazitätsplanung lediglich auf der Netzwerk- bzw. Transportschicht und berücksichtigen keine Anwendungen. Die Schwerpunkte des Verfahrens *Network Resource Planning (NRP)* nach [Make98] zur Analyse neuer lastkritischer Anwendungen liegen in der Dokumentation der Topologie und der Lasterhebung mit anschließender Lastvoraussage. Performance-Voraussagen werden abschließend mit Hilfe von Simulation gewonnen. Allerdings fehlt ein Hinweis auf die Erstellung und Validation der Simulationsmodelle. Positiv zu vermerken ist die Existenz einer Design- und Implementierungsphase, in der die per Simulation gewonnenen Erkenntnisse berücksichtigt werden.

Ein weiteres Vorgehen bei der Netzplanung stellt das Modell von CACI [Caci99] vor. Genau genommen ist es kein Vorgehensmodell, da lediglich eine Abfolge von zu erstellenden Systemmodellen beschrieben wird, das jedes für sich durch ein Tool der Firma unterstützt wird. Es werden keinerlei Hinweise auf Techniken oder Vorgehensweisen gegeben, und Abläufe oberhalb der OSI-Schicht 3 bleiben unberücksichtigt.

In allen Ansätzen fehlen Verfahren und Strategien, wie ein systematischer Kapazitätsplanungsprozess in einem Unternehmen etabliert werden kann, und wie die vielfältigen Aufgaben bei der Planung heterogener Client/Server-Systeme verteilt und koordiniert werden können.

02.03 Das MAPKIT Vorgehensmodell

In diesem Abschnitt wird ein Vorgehensmodell vorgestellt, mit dessen Hilfe ein umfassendes Kapazitätsmanagement für heterogene Systeme in einem Unternehmen eingeführt und anschließend erfolgreich praktiziert werden kann.

Das Vorgehensmodell umfasst die Aktivitäten *Vorbereitung der Kapazitätsplanung und Festlegung der Rahmenbedingungen, Erfassung und Darstellung der Ist-Situation*, *Planung zukünftiger Szenarien* und *Performance-Prognose und Dimensionierung*, vgl. Abbildung 02-1.

Um für die Idee des Kapazitätsmanagements im jeweiligen Unternehmen Akzeptanz zu schaffen, müssen die Denkweisen und Ansätze des Kapazitätsmanagements verdeutlicht und die daraus erwachsenden Vorteile aufgezeigt werden. Die Bestimmung organisatorischer und technischer Rahmenbedingungen sowie die Erfassung von Unternehmenszielen sind Voraussetzungen für das Gelingen der nachfolgenden Phasen. Die hier festgelegten

Restriktionen bestimmen die Wahl der Prognosemethoden und die Zielrichtung der Planung.

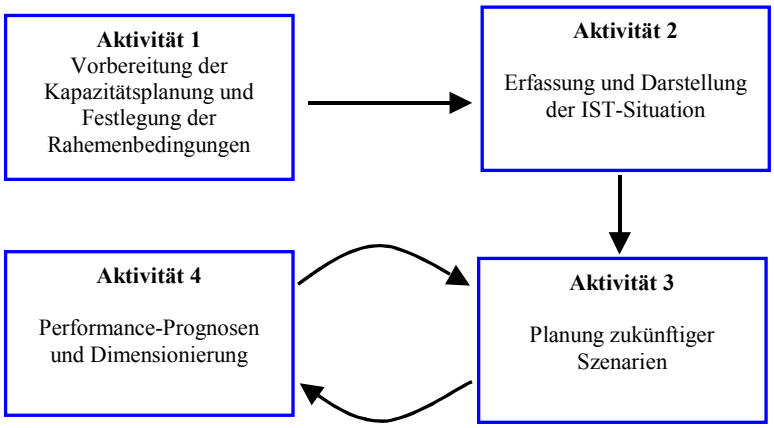


Abbildung 02-1: Phasen des MAPKIT-Vorgehensmodells

Aktivität 1 beinhaltet darüber hinaus die Benennung der zentralen Geschäftsprozesse des Unternehmens, inklusive ihrer Priorisierung sowie die Formulierung von Performance- und Verfügbarkeitsanforderungen aus Sicht der Geschäftsprozesse.

In Aktivität 2 wird eine Bestandsaufnahme des IT-Systems vorgenommen. Dazu werden alle im Unternehmen befindlichen HW-Komponenten wie Terminals, Server, Netze, Drucker etc. und die verwendete Software wie Betriebssysteme, Netzprotokolle und Anwendungen katalogisiert. Gleichzeitig wird die Topologie erfasst und in eine graphische Repräsentation überführt, um eine gemeinsame Kommunikationsbasis für das Unternehmen und das Planungsteam zu erstellen. Den zweiten Schwerpunkt in dieser Phase bildet die Erfassung von IT-Prozessen und die Abbildung von Geschäftsprozessen auf IT-Prozesse und IT-Komponenten, vgl. Abbildung 02-2. Weiterhin ist die auf das existierende System einwirkende Arbeitslast mit Hilfe von Hard- und Software-Monitoren zu erfassen und in ein geeignetes Lastmodell zu überführen. Das Lastmodell wird zusammen mit den Topologie-, Komponenten- und Prozessinformationen in ein Gesamtmodell der Ist-Situation überführt und mit Hilfe weiterer Messdaten kalibriert und validiert. Als Ergebnis entsteht ein Basismodell, das den Ausgangspunkt für spätere Prognosemodelle bildet. Falls das derzeitige IT-System keine Ausgangsbasis für die Prognose des zukünftigen Systems darstellt, können die Informationen zur Erstellung des Basismodells z.B. aus einer Teststellung oder einem Systemprototypen gewonnen werden.

Bevor nun Performance-Prognosen und Dimensionierungsvorschläge erstellt werden, ist im MAPKIT-Vorgehensmodell eine Planungsphase vorgesehen. In dieser Phase werden mögliche neue Anwendungen, neue Hardware-Komponenten oder neue Topologien systematisch erfasst. Für diese neuen Systeme werden Umfang und Ziele von Performance-Studien festgelegt.

Insbesondere kommt der Festlegung von Dienstgüte-Parametern für Anwendungen und/-oder Geschäftsprozessen eine zentrale Bedeutung zu. Die hier festgelegten Werte sind als Vorgaben zu verstehen, anhand derer die Ergebnisse der Prognosemodelle bewertet werden. Als letzte Aktivität beinhaltet das Vorgehensmodell eine Prognose- und Dimensionierungsphase. In das Basismodell sind gegebenenfalls Änderungen der Topologie bzw. Ergänzungen neuer Komponenten, die in der Systemplanungsphase festgelegt worden sind, zu integrieren. Neue Arbeitslasten sind ebenfalls zu prognostizieren, wobei hier je nach Art der neuen Arbeitslasten entsprechende Methoden ausgewählt werden.

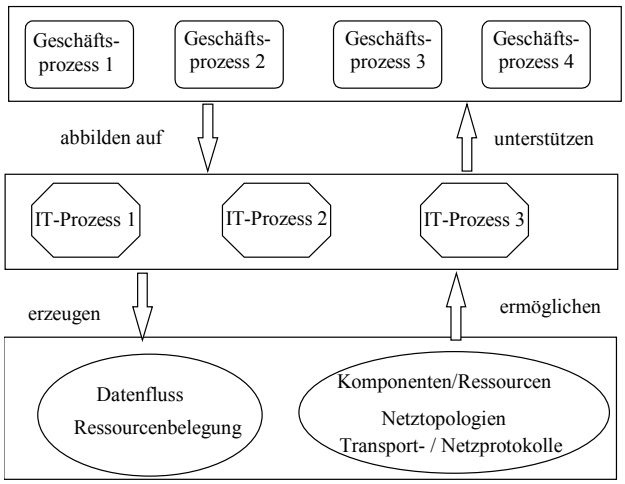


Abbildung 02-2: Beziehungen zwischen Geschäftsprozess, IT-Prozess und IT-Komponenten

Eine besondere Schwierigkeit stellt die Voraussage der Last dar, die durch völlig neue Anwendungen erzeugt wird. Die Ergebnisse des Prognosemodells werden mit den Performance-Anforderungen, die in der Planungsphase definiert wurden, verglichen, sieheAbbildung 02-3. Sollten die Modellergebnisse keine ausreichende Leistungsfähigkeit ergeben, müssen Systemparameter verändert werden. Sollte es nicht möglich sein, die erforderlichen Leistungsmerkmale zu erzielen, muss gegebenenfalls nochmals in die Planungsphase eingetreten werden, um den Systementwurf zu überarbeiten.

Im folgenden Kapitel wird nun anhand einer Studie gezeigt, wie das Vorgehensmodell in der Praxis angewendet werden kann.

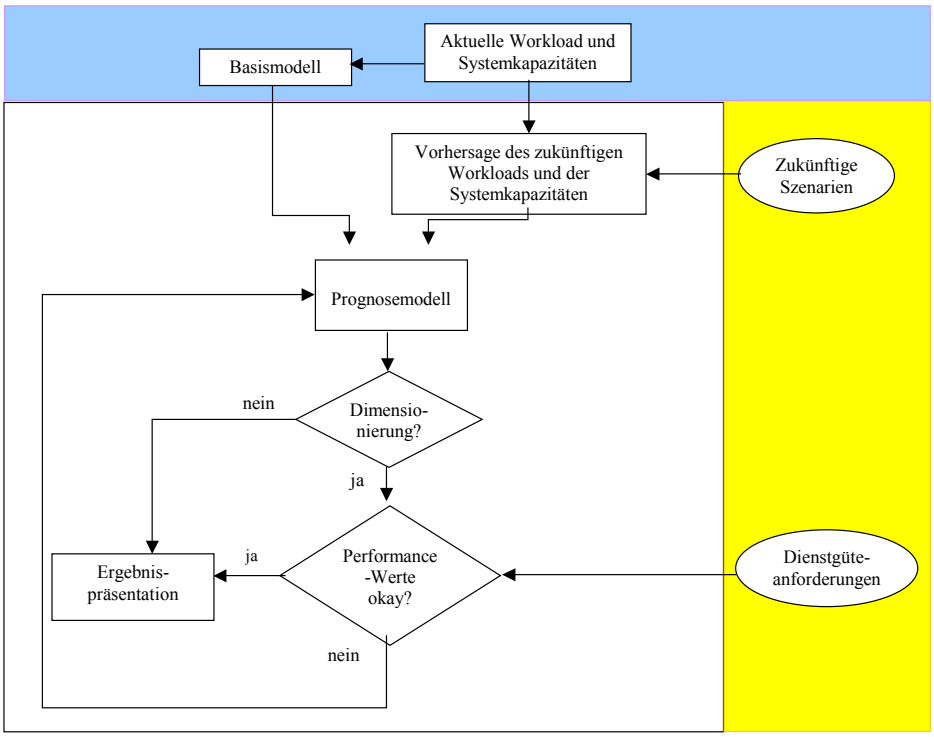


Abbildung 02-3: Aktivitäten der Performance-Prognose

02.04 Fallstudie zum MAPKIT Vorgehensmodell

Die Fallstudie untersucht eine große Dienstleistungsbehörde (im folgenden *DLB* genannt), die ein Mainframe-basiertes Anwendungssystem zu einem Client/Server-System migriert. Die *DLB* beschäftigt ca. 3500 Mitarbeiter und betreut mehrere Millionen Kunden.

Bei der Migration geht es nicht um eine komplette Ablösung des Mainframes, sondern um eine Erweiterung bzw. Modifizierung des bestehenden IT-Systems, das die eigentliche Unternehmensaufgabe der *DLB*, die Informationsverarbeitung und -bereitstellung, unterstützt. Es wird *Infosystem* genannt. Die Migration beinhaltet Veränderungen in den IT-Prozessen und den zugehörigen Hardware-Komponenten.

Die Studie beginnt zu einem Zeitpunkt, zu dem die ursprüngliche Mainframe-basierte Anwendung noch im Einsatz ist und die neue Anwendung in einem aus mehreren Rechnern bestehenden Teilsystem getestet wird. Das Ziel der Studie besteht darin, das IT-System für zukünftig zu erwartende Lasten zu dimensionieren.

(a) Vorbereitungen und Festlegung der Rahmenbedingungen

Zur Vorbereitung der Planungsstudie wird ein Workshop mit den zuständigen Mitarbeitern der *DLB* durchgeführt, bei dem eine einheitliche Begriffs- und Verständniswelt geschaffen wird. Weiterhin wird die Studie innerhalb der *DLB* motiviert und Methoden und Vorgehensweisen erläutert. Die Planungsziele werden abgesteckt und die Rahmenbedingungen für die Studie geklärt.

(b) Umfang und Zeitrahmen der Planungsstudie

Es muss die Möglichkeit bestehen, innerhalb eines kurzen Zeitraums umfangreiche Modellexperimente und die notwendigen Analysen durchzuführen. Die gesamte Studie inklusive ihrer Vorbereitung soll innerhalb eines Kalenderjahres abgeschlossen sein, da dann eine Entscheidung über die Einsatzform des neuen Systems ansteht.

(c) Technische Infrastruktur

Die *DLB* besitzt eigene Werkzeuge zum Monitoring ihres Systems in Form eines Hardware-Monitors und eines Netzwerk-Management-Tools. Performance-Modellierungswerkzeuge sind nicht vorhanden. Weiterhin sind Know-how und personelle Kapazitäten vorhanden, das System mit Hilfe der Werkzeuge zu vermessen und in eingeschränktem Umfang Erweiterungen der selbst entwickelten Anwendungs-Software vorzunehmen.

(d) Finanzielle Voraussetzungen

Die Kapazitätsplanung soll über den beschränkten Einsatz von Personen und Monitoring-Werkzeugen hinaus keine größeren Kosten verursachen.

(e) Geschäftsprozesse

Im Rahmen der zu untersuchenden Aktivitäten der *DLB* lassen sich im Wesentlichen zwei Kerngeschäftsprozesse ausmachen. Sie werden „Informationsabruf“ und „Informationseingabe“ genannt. Die entsprechenden Prozessketten sind in ihrer Grobstruktur in Abbildung 02-4 und Abbildung 02-5 dargestellt. Die Darstellung ist angelehnt an [Sche94]. Es wird unterschieden nach Ereignissen, die einen Prozess auslösen, der Funktion des Prozesses, der Organisationseinheit, die den Prozess durchführt, den Daten, die für den Prozess benötigt werden, und dem Typ des Prozesses.

Durch die Prozesskette „Informationsabruf“ erteilt ein Sachbearbeiter einem Kunden eine Auskunft, die dieser zuvor angefordert hat. In der Prozesskette „Informationseingabe“ nimmt ein Sachbearbeiter neue Kundeninformationen entgegen, die er in das Infosystem eingibt.

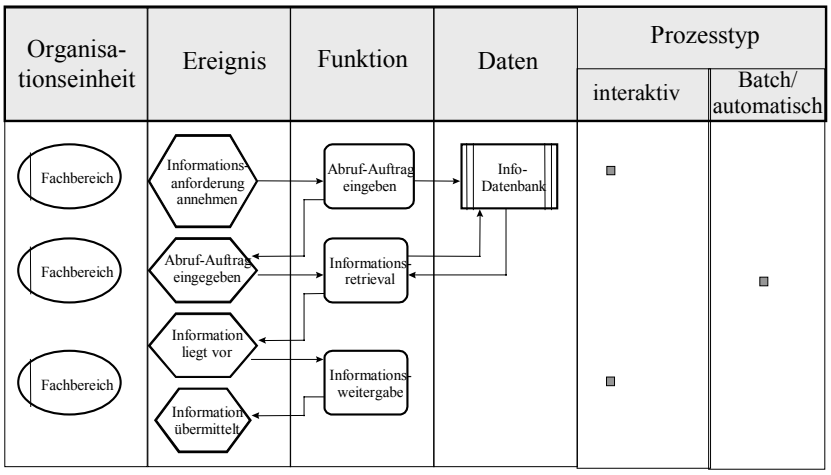


Abbildung 02-4: Prozesskette „Informationsabruf“

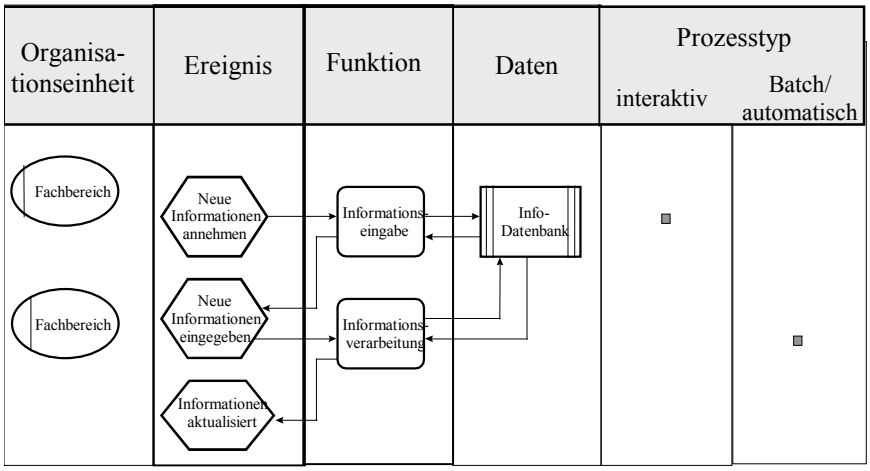


Abbildung 02-5: Prozesskette „Informationseingabe“

(f) Art der Anforderungen an das Zielsystem

Die Anforderungen an das Zielsystem werden in Form von Dienstgüteanforderungen auf IT-Systemebene angegeben. Sie werden in Mittelwerten ausgedrückt, die in einem stabilen System eingehalten werden sollen.

(g) Erfassung und Darstellung der Ist-Situation

Der Testbetrieb des neuen Infosystems stellt die Ist-Situation dar. Er eignet sich sehr gut zur Bildung eines Basismodells, da bei der Einrichtung der Teststellung möglichst viele Ähnlichkeiten mit dem zukünftigen System angestrebt werden und so bereits bedeutsame Parameter für Modelle zukünftiger Szenarien gewonnen werden können. Last und Topologie werden sich, ausgehend vom Testsegment, evolutionär entwickeln.

(h) Bestandsaufnahme

Das Infosystem wird innerhalb eines Ethernet-Segments mit 20 Benutzern betrieben. Abbildung 02-6 stellt die Topologie des IT-Systems graphisch dar. Die blass gezeichneten Komponenten gehören zum IT-System der *DLB*, auf sie wird im Testbetrieb aber nicht zugegriffen. Die 20 PCs der Teststellung befinden sich auf einer Etage und sind über ein 10BaseT-Ethernet an einen Etagen-Hub angeschlossen. Dieser ist durch ein 10BaseFL-Ethernet mit einem Switch im Rechenzentrum verbunden. Dort befindet sich ein FDDI-Ring mit einer Bandbreite von 100 MBits/s. An den Ring sind u.a. ein Host (*BS2000-Host*) und ein File-Server angeschlossen, der über das NFS-Protokoll angesprochen wird. Der FDDI-Ring wird ausschließlich im asynchronen Modus betrieben. Die Stationen am FDDI-Ring sind vom Typ DAS, d.h. sie sind als sog. Double Attached Station an beide Ringe angeschlossen (siehe [FDDI86]).

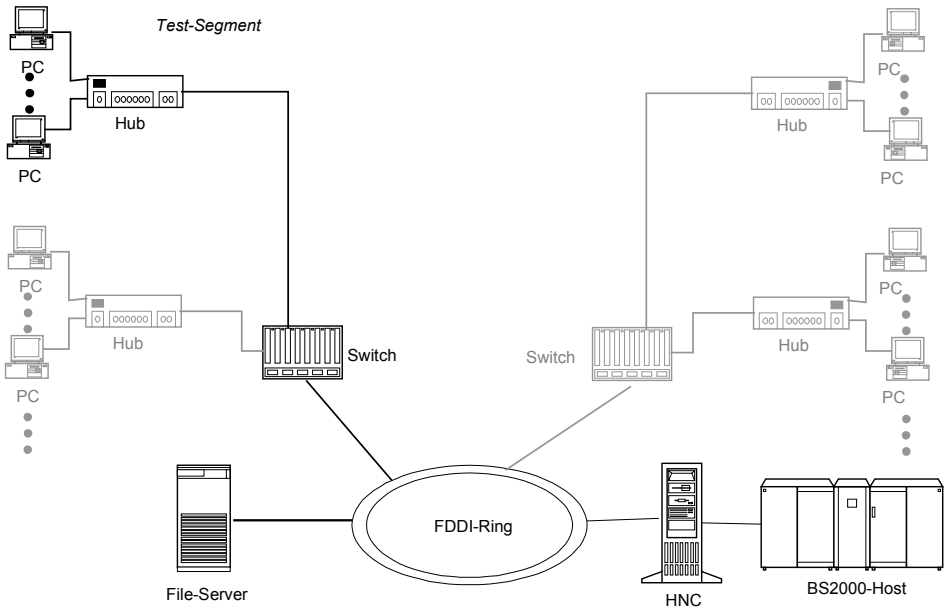


Abbildung 02-6: Topologie des IT-Systems

Die Kapazitäten der Systemkomponenten wirdn in Tabelle 02-1 zusammenfassend dargestellt.

Komponente	Kapazität
PC	nicht bekannt
Ethernet	10 MBit/s
FDDI-Ring	100 MBit/s
File-Server	2 MByte/s
BS2000-Host	nicht bekannt

Tabelle 02-1: Kapazität der Systemkomponenten

Angaben über die Kapazität der PCs und des BS2000-Hosts, die sich auf die Unterstützung des Infosystems beziehen, sind bei der *DLB* nicht zu gewinnen. Bei der Erstellung des Basismodells muss eine entsprechende Problemlösung gefunden werden.

(i) IT-Prozesse und beteiligte Komponenten

Die für den Kapazitätsplanungsauftrag relevanten IT-Prozesse werden von einer Anwendung namens *GRUI+* bereitgestellt. *GRUI+* setzt sich aus zwei Komponenten zusammen. Die eine Komponente - *TEUI* genannt - realisiert die eigentliche Anwendungslogik und läuft auf dem BS2000-Host.

Die zweite Komponente namens *GRUI* realisiert die graphische Oberfläche der Anwendung. Die hierfür notwendigen Daten werden von dem File-Server geladen und anschließend vom PC zur graphischen Aufbereitung der *TEUI*-Daten verarbeitet. Die GRUI+-Anwendung läuft folgendermaßen ab:

Beim Start der Anwendung wird ein Basismodul mit Oberflächen-(GUI-)Daten vom File-Server zum PC am Arbeitsplatz des Benutzers übertragen. Dieses Modul wird *Startsegment* und die Phase der Anwendung *Startphase* genannt. Anschließend geht die Anwendung in die sogenannte *Betriebsphase* über: Anfragen und Datenzugriffe werden in Form von *TEUI*-Transaktionen an den BS2000-Host gestellt. Die vom Host gelieferten Nutzdaten werden mit Hilfe der im PC vorhandenen GUI-Steuerdaten dargestellt. Während der Betriebsphase kann es vorkommen, dass einige GUI-Module vom File-Server nachgeladen werden müssen, wenn das Startsegment zur Nutzdatendarstellung nicht ausreicht.

(j) Abbildung der Geschäftsprozesse auf IT-Prozesse

Die Geschäftsprozesse „Informationsabruf“ und „Informationseingabe“ werden beide durch die Anwendung *GRUI+* unterstützt.

(k) Lastmodell

Die Gewinnung von Daten über die Last- und Verkehrscharakteristik stellte das eigentliche Problem der Fallstudie dar. Versuche, die von der Anwendung erzeugte Verkehrslast im „leeren“ System mittels Hardware-Monitoren zu messen, schlugen fehl. Die Ursachen hierfür lagen zum einen in Verständigungsschwierigkeiten zwischen dem MAPKIT-Team und der *DLB*, zum anderen in der fehlenden Möglichkeit, bei der *DLB* zu irgendeinem Zeitpunkt ein völlig unbelastetes System vorzufinden. Das Monitoring in einem Test-Labor, in dem das IT-System und die Anwendung nachgebaut werden sollten, war ebenfalls nicht möglich, da einige Leistungsgrößen des Systems für den korrekten Nachbau nicht zu ermitteln waren.

Man entschied sich für eine clientseitige Transaktionsverfolgung im laufenden Betrieb der Anwendung. So konnten Informationen über das Benutzerverhalten und über die Ressourcen-Anforderungen der Transaktionen gewonnen werden. Hierfür wurde die Anwendung *GRUI+* erweitert: Sie wurde mit Trace-Punkten versehen, wodurch bestimmte Ereignisse

innerhalb der Anwendung auf der Client-Seite festgehalten werden. *GRUI+* erzeugt pro PC, auf dem es ausgeführt wird, einen „Event-Trace“. Ein Auswerteprogramm liest die Trace-Dateien, wertet sie aus und erzeugt Statistiken über die Last- und Verkehrscharakteristik, aus denen schließlich Parameter für die Modellierung gewonnen werden, vgl. Abbildung 02-7.

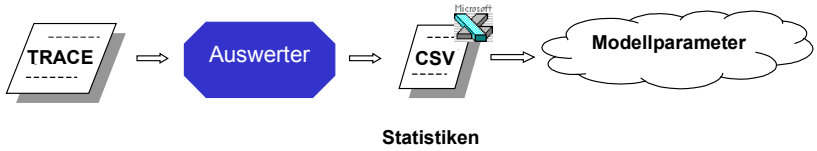


Abbildung 02-7: Gewinnung von Last- und Verkehrsprofilen

Die innerhalb von *GRUI+* auftretenden Aufträge lassen sich in unterschiedliche Klassen einteilen. Aufträge einer Klasse durchlaufen jeweils dieselben Systemkomponenten und belasten diese mit dem gleichen Lastvolumen. Es können folgende Lastklassen gebildet werden:

- Startphase (Laden des Startsegments vom File-Server)
- Modul Nachladen (Laden eines Moduls vom File-Server)
- Transaktion (Ausführen einer Transaktion ohne Modul-Laden)

Mittlere Datenvolumina		
	Startsegment	**1 MByte**
	Nachgeladenes Modul	**200 KByte**
	Transaktion der Betriebsphase	**2 KByte**
Benutzerdenkzeit		**60 Sekunden**
Verhältnis **Transaktion: Modul-Nachladen** in der Betriebsphase		**4 : 1**

Tabelle 02-2: Last- und Verkehrscharakteristik

Durch den Auswerter sind die Datenvolumina bzw. Verweilzeiten zu ermitteln, mit der Aufträge einer Lastklasse die Systemressourcen beanspruchen.

Tabelle 02-2 stellt die Werte für die Last- und Verkehrscharakteristik dar. Die Last wird also durch die Datenvolumen und die Häufigkeit der Auftragsanforderung beschrieben.

(I) Basismodell

Das Lastmodell und die Informationen aus der Bestandsaufnahme werden nun zu einem Basismodell zusammengeführt. Unter Berücksichtigung der gegebenen Rahmenbedingungen werden als Modellierungsmethode geschlossene Warteschlangennetze gewählt.

Die Analyse des Modells wird mit einem Tool namens *VITO* durchgeführt. Hierbei handelt es sich um ein Werkzeug zur mathematischen, approximativen Analyse von geschlossenen Warteschlangennetzwerken (siehe [VITO99] und [VITOHp]). Folgende Gründe führten zur Auswahl der Modellierungsmethode und des Werkzeugs:

- *VITO* ist ein analytisches Modellierungswerkzeug mit einem effizienten approximativen Algorithmus, wodurch Modellexperimente sehr kurze Laufzeiten aufweisen.
- *VITO* ist eine Eigenentwicklung innerhalb des MAPKIT-Projekts, d.h. durch den Einsatz des Werkzeugs fallen für die *DLB* keine Lizenzgebühren an.
- *VITO* führt eine stationäre Mittelwertanalyse durch, d.h. die Art der Modellergebnisse stimmt mit den Dienstgüteanforderungen überein, wodurch sich die Einhaltung dieser Werte über die Modelle überprüfen lässt.

Abbildung 02-8 stellt das Basismodell in VITO-Notation graphisch dar.

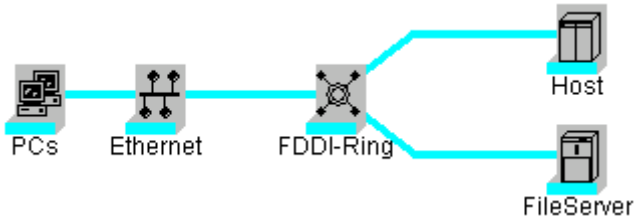


Abbildung 02-8: Das Basismodell in *VITO*-Notation

Die Station *PCs* stellt die 20 Arbeitsplatz-PCs dar, die durch einen Hub zu einem Ethernet-Segment zusammengeschlossen werden. Das Verhalten des Hubs wird durch die Station *Ethernet* mitberücksichtigt. Das Ethernet ist mit einem Switch verbunden, der es an den FDDI-Ring im Rechenzentrum anschließt. Da der Switch keine signifikanten Auswirkungen auf die System-Performance hat, wird lediglich der FDDI-Ring als Station modelliert. Im Modell werden weiterhin der *BS2000-Host* und der *File-Server* abgebildet.

Der *BS2000-Host* und die *PCs* werden als *Infinite Server* (IS) modelliert, da zwar ihre Kapazitäten unbekannt sind, aber die Verweilzeiten von Aufträgen an ihnen entweder bekannt sind oder geschätzt werden können.

Ethernet und *FDDI-Ring* werden als *Queue Dependend Server* (QD) modelliert, da ihre Effizienz sich mit zunehmender Auftragsanzahl verändert. Während sie beim Ethernet wegen der zunehmenden Anzahl von Kollisionen bei wachsender Anzahl sendebereiter Stationen sinkt, steigt sie beim FDDI-Ring aufgrund der optimaleren Auslastung des Mediums. Der *File-Server* wird als *Processor Sharing*-Station (PS) modelliert.

Die Parametereingabe erfolgt bei *VITO* in Form von *Bedienwunsch* (r), *Stationsgeschwindigkeit* (v) und *Besuchshäufigkeit* (h). Der Bedienwunsch ergibt sich aus dem übertragenen Datenvolumen (siehe Lastmodell) oder aus der ermittelten/geschätzten Verweilzeit, die Stationsgeschwindigkeit aus der Stationskapazität. Die Besuchshäufigkeit beschreibt die Anzahl des Eintreffens eines Auftrags einer bestimmten Lastklasse an einer Station.

Die Lastklassen werden bei *VITO* mit Hilfe von *Ketten* modelliert. Eine Kette wird jeweils von einer bestimmten Anzahl von Aufträgen durchlaufen. Die Anzahl von Aufträgen wird *Population* (p) genannt.

Es werden für das Basismodell entsprechend der Lastklassen Ketten mit den folgenden Namen gebildet:

1. Startphase
2. Modul nachladen
3. Transaktion

Der Durchsatz der Ketten 2 und 3 wird mit Hilfe von *VITO* so gekoppelt, dass im Mittel für vier ausgeführte Transaktionen ein Modul nachgeladen wird, vgl.Tabelle 02-2.

Tabelle 02-3 stellt eine Übersicht der Modell-Eingabeparameter dar.

Stationen / Ketten	PCs r	PCs h	Ethernet (v = 10 MBit/s) r	Ethernet h	FDDI (v = 100 MBit/s) r	FDDI h	BS2000-Host r	BS2000-Host h	File-Server (v = 2 MByte/s) r	File-Server h
Startphase (p = 20)	8 s	1	1 MByte	1	1 MByte	1	--	--	1 MByte	1
Modul nachladen (p = 20)	1,6 s	1	200 KByte	1	200 KByte	1	--	--	200 KByte	1
Transaktion (p = 20)	61,4 s [2]	1	2 KByte	1	2 KByte	1	0,06 s	1	--	--

Tabelle 02-3: Modell-Eingabeparameter

Das so modellierte System wird mit *VITO* analysiert. Da die Analyse des Basismodells der Validation und der Kalibrierung der Eingangsparameter dient, werden an dieser Stelle nur die Ergebniswerte betrachtet, die sich validieren lassen. Dabei handelt es sich um die *GRUI+*-Antwortzeiten. Abbildung 02-9 stellt das Ergebnis der Modellexperimente graphisch dar.

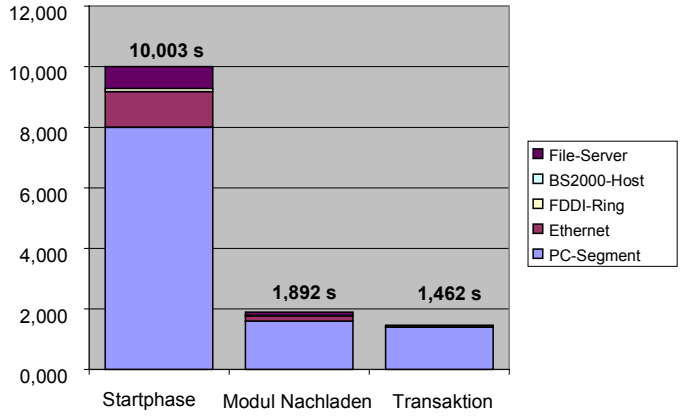


Abbildung 02-9: Mittlere Antwortzeiten und deren Zusammensetzung

[2] Im Bedienwunsch der PCs sind 60 sec Denkzeit der Benutzer und 1,4 sec PC-Verarbeitungszeit enthalten.

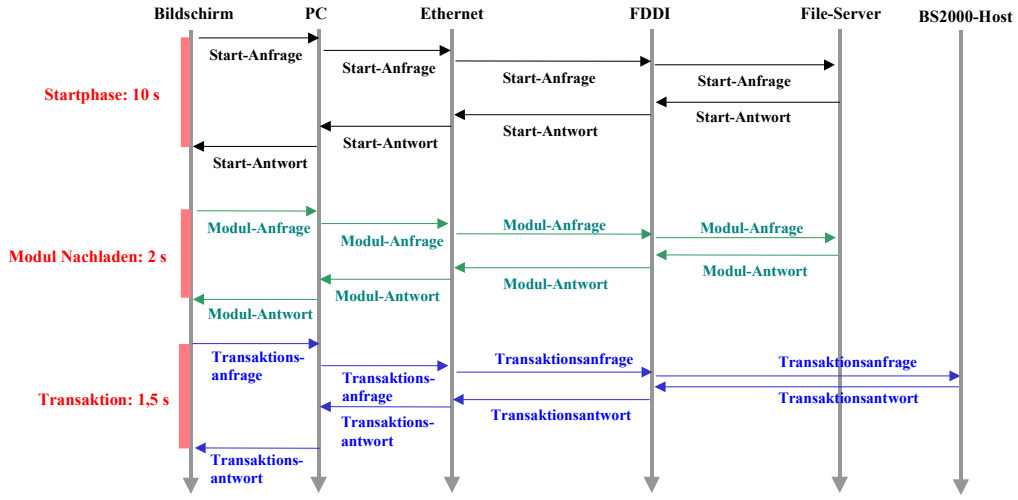


Abbildung 02-10: *GRUI+*-Antwortzeiten als Weg/Zeit-Diagramm

Durch das im Kapitel „Lastmodell“ beschriebene Tracing wurden zum einen Werte für das Lastmodell (Benutzerverhalten, Verarbeitungszeiten, Datenvolumen) ermittelt, zum anderen wurden Antwortzeiten von *GRUI+* gewonnen, die zur Modellvalidation dienen. Beim Vergleich der durch das Modell ermittelten Antwortzeiten und den durch Messung ermittelten Werten ist festzustellen, dass die Abweichungen nur wenige Prozent betragen, z.B. 1.46 sec statt 1.4 sec für die mittlere Transaktionsantwortzeit bzw. 1.89 sec statt 1.6 sec für das Nachladen eines Moduls.

In dem Modell wurden keine weiteren Anwendungen berücksichtigt, da diese das Systemverhalten nur unwesentlich beeinflussen. Hätte das Basismodell größere Abweichungen zu den Messungen aufgewiesen, wäre die Modellierung einer Hintergrundlast ein Ansatzpunkt für die Kalibrierung.

02.05 Planung zukünftiger Szenarien

Die Anwendung *GRUI+* soll zukünftig in der Form bestehen bleiben, in der sie im Testsegment betrieben wird. Es ist geplant, alle Arbeitsplätze mit dieser Software auszustatten. Daher müssen sowohl zusätzliche Komponenten des IT-Systems als auch höhere Arbeitslasten berücksichtigt werden. Das Basismodell eignet sich so als Grundlage für Modelle der zukünftigen Szenarien, da sich diese aus Erweiterungen bzw. Modifikationen des Ausgangssystems ergeben.

Zunächst ist ein Szenario zu entwickeln, in dem das gesamte IT-System mit der *GRUI+*-Last mehrerer Anwender belastet wird. In das Prognosemodell müssen somit die über das Testsegment hinausgehenden Systemkomponenten integriert werden. Die Anzahl aktiver Benutzer kann in verschiedenen Szenarien variiert werden. Es sind Extremfälle zu berücksichtigen, bei denen sehr viele bzw. sehr wenige Benutzer das System belasten.

Weiterhin erwägt die *DLB*, einen zusätzlichen FDDI-Ring speziell für *GRUI+* einzusetzen. Dieses Szenario ist ebenfalls zu berücksichtigen. Tabelle 02-4 stellt die zu betrachtenden Szenarien übersichtsartig dar.

Systemkonfiguration / Last	Geringe Last	Mittlere Last	Hohe Last
Gesamtsystem in bisheriger Form	✓	✓	✓
Gesamtsystem mit zusätzlichem FDDI-Ring	✓	✓	✓

Tabelle 02-4: Zukünftige Szenarien

(a) Dienstgüten

Für die *DLB* ist es wichtig, bestimmte mittlere Antwortzeiten für *GRUI+* nicht zu überschreiten. Weitere Dienstgüteanforderungen ergeben sich von technischer Seite für das Ethernet und den File-Server: Die Auslastung des Ethernets sollte 30 % und die mittlere Auslastung des File-Servers über einen längeren Zeitraum sollte 85 % nicht überschreiten, um eine reibungslose Funktionsweise der Komponenten zu ermöglichen. Im einzelnen ergeben sich folgende Dienstgüteanforderungen:

Benutzerbezogene Dienstgüteanforderungen	
Mittlere Antwortzeit in der Startphase	≤ 10 s
Mittlere Antwortzeit beim Modul Nachladen	≤ 3 s
Mittlere Antwortzeit einer Transaktion in der Betriebsphase ohne Modul nachzuladen	≤ 2 s
Systembezogene Dienstgüteanforderungen	
Mittlere Auslastung Ethernet	≤ 30 %
Mittlere Auslastung File-Server	≤ 85 %

Tabelle 02-5: Leistungsmaße und Dienstgüten

02.06 Performance-Prognosen und Dimensionierung

(a) Gesamtsystem mit einem FDDI-Ring

Zunächst werden die Szenarien untersucht, die das gesamte IT-System mit nur einem FDDI-Ring betrachten. Das System ist in einem Gebäude verteilt auf 27 Etagen mit jeweils 30 Benutzern. Die Topologie erweitert sich somit um 26 zusätzliche Etagen.

Die Verkehrslast, die durch einen einzelnen Anwender erzeugt wird, wird aus dem Basismodell übernommen. Das zu betrachtende System wird durch die Abbildung 02-6 inklusive der blass gezeichneten Komponenten beschrieben.

Abbildung 02-11 stellt das resultierende Warteschlangennetz in *VITO*-Notation graphisch dar.

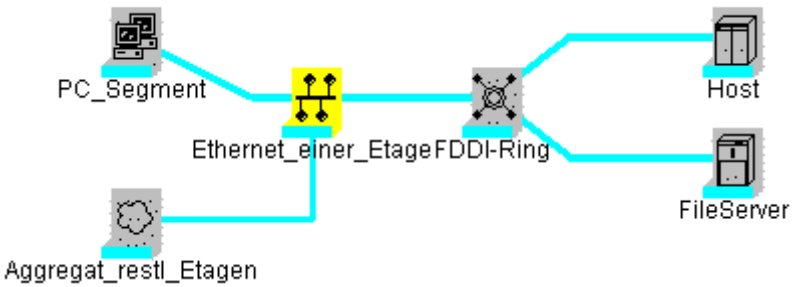


Abbildung 02-11: Das Basismodell mit agreggiertem Teilmodell

Die Stationen *PC-Segment* und *Ethernet_einer_Etage* stellen eine Etage des Verwaltungsgebäudes dar. Die PCs und Ethernet-Segmente der restlichen Etagen werden aggregiert und als eine Station modelliert, da nicht die Performance jeder einzelnen Etage betrachtet werden muss, sondern es ausreicht, eine typische Etage durch die Stationen PC-Segment und Ethernet zu modellieren.

Die Eingabeparameter werden zum großen Teil aus dem Basismodell übernommen. Zusätzlich kommen Ketten für die aggregierte Komponente hinzu.

Die Aggregat-Station wurde mit Hilfe einer sog. Kurzschlussanalyse parametrisiert, vgl. Tabelle 02-6.

Ketten/Stationen für das Aggregat	Aggregat-Station		FDDI (v = 100 MBit/s)		BS2000-Host		File-Server (v = 2 MByte/s)	
	r	h	r	h	r	h	r	h
Startphase	8 s	1	1 MByte	1	--	--	1 MByte	1
Modul Nachladen	1,6 s	1	200 KByte	1	--	--	200 KByte	1
Transaktion	61,4 s	1	2 KByte	1	0,06 s	1	--	--

Tabelle 02-6: Modell-Eingabeparameter bzgl. der aggregierten Etagen

Die Populationen der einzelnen Ketten werden für die verschiedenen Szenarien „geringe Systembelastung", „mittlere Systembelastung" und „hohe Systembelastung" variiert. Die Populationen für diese Szenarien ergeben sich aus Beobachtungen des Betriebs des Text-basierten Vorgängersystems von *GRUI+*, da der Benutzerkreis der gleiche geblieben ist.

Geringe Systembelastung: In diesem Szenario wird der Fall betrachtet, dass 500 Benutzer *GRUI+* ausführen und davon 10 Benutzer innerhalb von 5 Minuten *GRUI+* neu starten.

Mittlere Systembelastung: In diesem Szenario wird der Fall betrachtet, dass 600 Benutzer *GRUI+* ausführen und davon 30 Benutzer innerhalb von 5 Minuten *GRUI+* neu starten.

Hohe Systembelastung: Es wird das Szenario modelliert, dass 700 Benutzer *GRUI+* ausführen und alle innerhalb von 10 Minuten die Anwendung neu starten. Diese Situation entsteht typischerweise nach einem Systemausfall, wenn alle Benutzer neu booten müssen. Tabelle 02-7 stellt die Populationen der einzelnen Ketten in den verschiedenen Szenarien dar.

Kette / Szenario	Geringe Last	Mittlere Last	Hohe Last
Für die einzelne Etage			
Startphase	5	15	30
Modul Nachladen	30	30	30
Transaktion	30	30	30
Für das Aggregat			
Startphase	5	15	670
Modul Nachladen	470	570	670
Transaktion	470	570	670

Tabelle 02-7: Populationen für die Lastketten

Die Analyse der einzelnen Szenarien zeigt, dass die vorgegebenen Dienstgüten (siehe Tabelle 02-5) zwar bei geringer und mittlerer Systembelastung eingehalten werden können, dass sie allerdings im Hochlastfall weit überschritten werden, vgl. Abbildung 02-12 und Abbildung 02-13.

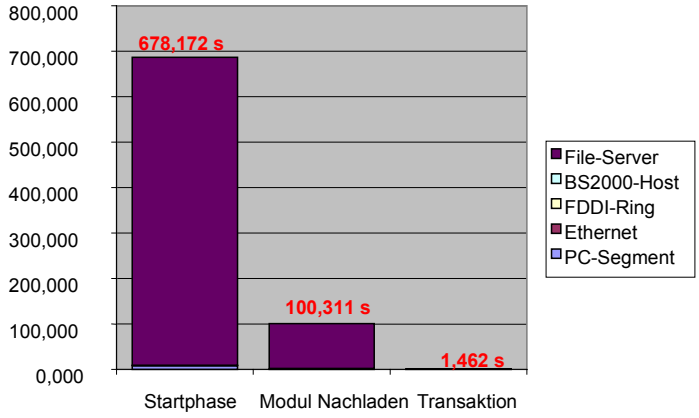


Abbildung 02-12: Antwortzeiten unter hoher Systembelastung

Die Antwortzeit in der Startphase verletzt die geforderte Dienstgüte um über 6600 %, beim Nachladen von Modulen beträgt die Differenz über 3200 %. Diese Werte sind völlig inakzeptabel. Die vorgegebene Dienstgüte bzgl. des Ethernets wird eingehalten, wohingegen die des File-Servers deutlich überschritten wird. Die mittlere Auslastung von nahezu 100 % verursacht die viel zu hohen Antwortzeiten, was auch daran zu erkennen ist, dass die Antwortzeit in der Startphase und beim Modul Nachladen zum größten Teil aus der Verweilzeit beim File-Server besteht.

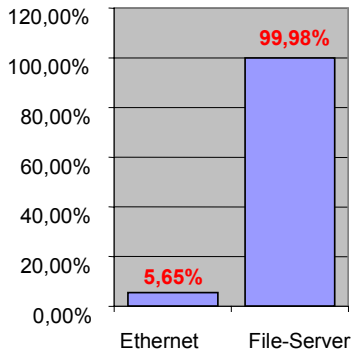


Abbildung 02-13: Auslastung von Ethernet und File-Server

Diese Ergebnisse führen zu folgendem neuen Szenario, das bisher noch nicht berücksichtigt wurde:

Hohe Systembelastung bei zwei File-Servern: In das Modell wird ein zusätzlicher File-Server eingebaut, der das gleiche Leistungsvermögen wie der bisherige File-Server besitzt. Die übrigen Modellparameter bleiben unverändert. Die Analyse des Modells ergibt, dass in diesem Szenario die geforderten Dienstgüten eingehalten werden.

(b) Gesamtsystem mit zusätzlichem FDDI-Ring

Während sich durch die Analyse der ersten Szenarien ein neues zu betrachtendes Szenario ergibt, verliert der Einsatz eines zusätzlichen FDDI-Rings speziell für *GRUI+* an Bedeutung.

Selbst in den Szenarien mit hoher Systembelastung ist der einzelne FDDI-Ring maximal zu 15 % ausgelastet, so dass sich aus der Sicht von *GRUI+* keine Notwendigkeit für einen zweiten Ring ergibt. Die Modellierung und Analyse bestätigt diese Annahme; d.h. ein zusätzlicher FDDI-Ring wäre allenfalls zu empfehlen, wenn mit einer sehr hohen Hintergrundlast zu rechnen ist.

(c) Auswirkungen auf Geschäftsprozesse und Dimensionierungsempfehlung

Würde *GRUI+* in das zur Zeit bestehende IT-System übernommen, wären die Geschäftsprozesse „Informationsabruf“ und „Informationseingabe“ nicht mehr performant durchführbar, da es durch die schlechten Antwortzeiten der Anwendung zu unverhältnismäßigen Leerzeiten innerhalb der Prozesskette käme. Die Ausführung der Geschäftsprozesse durch

die Sachbearbeiter wäre nicht mehr zumutbar. Die Einführung eines zusätzlichen File-Servers ermöglicht einen performanten Betrieb. Die Kosten für einen zweiten FDDI-Ring können eingespart werden.

02.07 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Vorgehensmodell vorgestellt, das den gesamten Kapazitätsplanungsvorgang für IT-Systeme umfasst. Die Verwendbarkeit des Vorgehensmodells in der Praxis wurde anhand einer Fallstudie demonstriert. Es wird für die Zukunft angestrebt, den Ansatz anhand von weiteren Dimensionierungsstudien weiter zu entwickeln und zu etablieren. Insbesondere ist die Eignung in stärker verteilten und heterogenen Systemen zu erproben.

02.08 Referenzen

[FDDI86] FDDI Token Ring Media Access Control MAC. ANSI X3T9.5, 1986.

[Jain90] R. Jain: Performance Analysis of FDDI Token Ring Networks: Effect of Parameters and Guidelines for Setting TTRT. Modifizierte Version des Papiers von Proc. ACM SIGCOMM'90, Philadelphia, September 1990.

[Jain91] R. Jain: The Art of Computer Systems Performance Analysis. John Wiley & Sons, Inc., New York 1991.

[Lazo84] E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik: Quantitative System Performance. Prentice-Hall, Inc., New Jersey 1984.

[Make98] http://www.makesystems.com/Pages/CORPnewsclips8make.html

[Mena94] D. A. Menascé, V. A. F. Almeida, L. W. Dowdy: Capacity Planning and Performance Modelling; From Mainframes To Client-Server Systems. Prentice-Hall PTR, Englewood Cliffs, New Jersey 1994.

[Sche94] A.-W. Scheer: Business Process Engineering. Springer-Verlag, Berlin 1994.

[Sche90] A.-W. Scheer: Wirtschaftsinformatik. Springer-Verlag, Berlin 1990.

[Caci99] http://www.caciasl.com

[VITO99] M. Vilents, C. Flüs, B. Müller-Clostermann: VITO: A Tool for Capacity Planning of Computer Systems. MMB'99, 10. GI/ITG-Fachtagung 22.-24.09.1999 in Trier. Forschungsbericht Nr. 99-17, Kurzbeiträge und Toolbeschreibungen.

[VITOHp] VITO-Homepage: http://www.cs.uni-essen.de/VITO.

KAPITEL 03 E-MANAGEMENTLEITSTAND "LIGHT"

ACHIM MANZ-BOTHE, MATHIAS HEHN UND FRANZ-JOSEF STEWING

03.01 Motivation und Einführung

Durch die derzeit stattfindende Rezentralisierung und Konsolidierung entsteht in den Rechenzentren, Serverfarmen und Application Service Providern ein Management- und Überwachungsaufwand, der mit klassischen Tools hohen Personal- und Qualifizierungsbedarf nach sich zieht.

Proprietäre, meist nicht in vorhandene Infrastrukturen zu integrierende Einzellösungen verursachen hohe Hardware-, Software- und Schulungskosten. Durch plattformspezifische Lösungen koexistieren häufig unterschiedliche Managementplattformen nebeneinander. Hardwarebeschaffungen unterliegen nicht der Firmenpolitik, sondern der jeweiligen Spezifikation der einzusetzenden Managementplattform. Weiterhin führen marktübliche Managementsysteme zu einem hohen Installationsaufwand. Das Management der Managementsysteme wird seinerseits zum bestimmenden Faktor. Die Bewältigung eines Ausfalls des Managementsystems ist oft nicht aus eigenen Kräften möglich, da ggf. Hardware oder Installations-Know-how fehlen.

Die Leitstände der Rechenzentren und Provider werden immer komplizierter. Falsch interpretierte oder aufgrund der Anzeigenflut nicht beachtete Anzeigen bedeuten Kosten, Ausfallzeiten und den Verstoß gegen Service Level Agreements. Die für den unmittelbaren Rechnerbetrieb Verantwortlichen müssen eine immer höhere Qualifikation aufweisen; die Grenze zum Second Level Support verwischt.

03.02 Aufgabenstellung

Ein an die Leitstände kritischer Systeme der klassischen Industrie angelehntes Managementsystem muss sich an den Bedürfnissen des überwachenden Personals ausrichten. Die vielfach äußerst weitreichenden Eingriffsmöglichkeiten gängiger Managementsysteme sind für den Überblick über den Gesamtprozess regelmäßig nicht erforderlich und auch nicht sinnvoll.

Vielmehr soll das System dem Betreiber einen klaren, Fehlinterpretationen vorbeugenden Eindruck vom Zustand der vielfältigen Betriebsparameter vermitteln. Die Managementclients selbst sollen plattformunabhängig sein und damit die Möglichkeit bieten, auf den bereits existierenden Arbeitsplatzrechnern abzulaufen.

Jedem Mitarbeiter soll genau die Menge von Informationen dargestellt werden, die in seinen Verantwortungsbereich fällt. Die Konfiguration der dargestellten Informationen soll einfach und auch vom Nicht-Spezialisten durchführbar sein. Eine aufwändige Schulung soll für den täglichen Betrieb, aber auch für die Konfiguration des Systems, nicht erforderlich sein. Auf der Serverseite sollen vorhandene Systeme weiter einsetzbar sein. Niedrige Hardwareanforderungen halten das Verhältnis der Kosten des Managements zu denen der zu überwachenden Produktivsysteme im richtigen Verhältnis.

Das System soll stabil und ausfallsicher sein und die überwachten Systeme nicht nennenswert belasten. Gleichzeitig soll es von den „kleinsten" Installationen bis hin zu firmenweiten Überwachungslösungen skalier- und erweiterbar sein. Ein offenes Konzept soll die Integration zusätzlicher Features sowie ein Wachsen des Systems sicherstellen.

03.03 Technisches Konzept

Die identifizierten Anforderungen „Plattformunabhängigkeit" und „einfache Installation" legen die Verwendung einer web-zentrierten Architektur nahe (Abbildung 03-1). Clientseitig stellen HTML-Seiten mit eingebetteten Java-Applets die zu visualisierenden Informationen dar. Durch den Verzicht auf RMI-Kommunikation und Rückgriff auf BSD-Sockets sind auch solche Webbrowser verwendbar, deren Hersteller die Integration dieses Elements des Java-Sprachstandards nicht für nötig erachtet haben. Durch die Kapselung der Kommunikation ist ein Umstieg auf die Verwendung von RMI jederzeit mit wenig Aufwand möglich, sobald ausschließlich geeignete Webbrowser eingesetzt werden.

Durch die Entscheidung zum Einsatz von HTML und Java ist automatisch jeder Standardarbeitsplatz als Einsatzort des Leitstandes vorbereitet. Die weitergehende Installation reduziert sich auf das Mitteilen einer URL. Die auf der Clientseite eingeführte Flexibilität soll auf der Serverseite weitergeführt werden. Daher kommt dort ebenfalls Java zum Einsatz. Für jeden überwachten Systemparameter wird ein Modul eingesetzt. Die Module sind vollständig unabhängig voneinander und können getrennt voneinander eingesetzt, erweitert und konfiguriert werden.

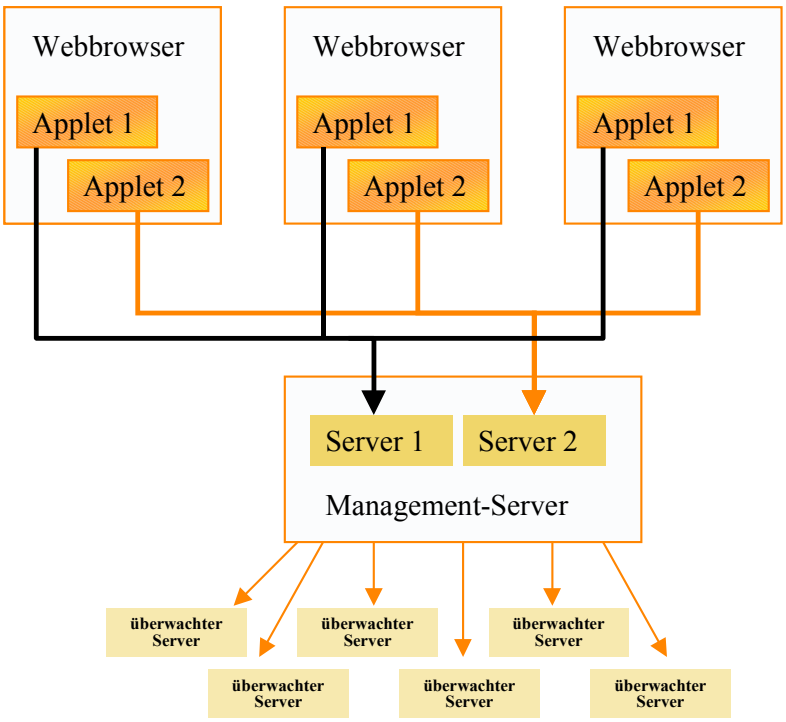


Abbildung 03-1: Architekturmodell

Jedes Modul läuft auf dem oder den Managementserver(n) als eigenständiger Prozess. Durch diese Konstruktion wird nicht nur ein Multiprozessorsystem als Managementplattform optimal genutzt, auch die Verteilung der Module auf mehrere Managementsysteme bietet sich an. Ein Failover-Konzept des Managementsystems ist durch eine Erweiterung der Clients jederzeit möglich.

03.04 Technische Realisierung

Das vorgestellte Architekturmodell wurde in Form eines Prototypen realisiert. Eine Menge von Java-Applets greift mithilfe von TCP/IP-Verbindungen auf eine Menge von Serverprozessen zu. Hierbei wird der zu überwachende Server dem Managementprozess als Parameter mitgegeben. Dieser erfasst mittels rsh-Aufrufen den angefragten Betriebsparameter und liefert ihn als Zahlenwert zurück. Die Visualisierung erfolgt durch das Applet (Abbildung 03-2).

Die Konfiguration des Systems erfolgt in der HTML-Seite. Mit Hilfe von übergebenen Parametern wird den Applets mitgeteilt, welcher Server in welchen zeitlichen Abständen überwacht werden soll. Auch die Architektur des Servers wird als Parameter mitgeteilt, um unterschiedliche Algorithmen und Messverfahren für verschiedene Rechnersysteme zu gestatten.

Die Serverprozesse selbst bieten ihre Dienste an fest definierten Ports an. Jeder Server kann einen speziellen Betriebsparameter überwachen.

Die erfassten Werte werden je nach Wichtigkeit unterschiedlich eingefärbt. Ausfälle und langfristige Entwicklungen werden visuell erfassbar (Abbildung 03-3).

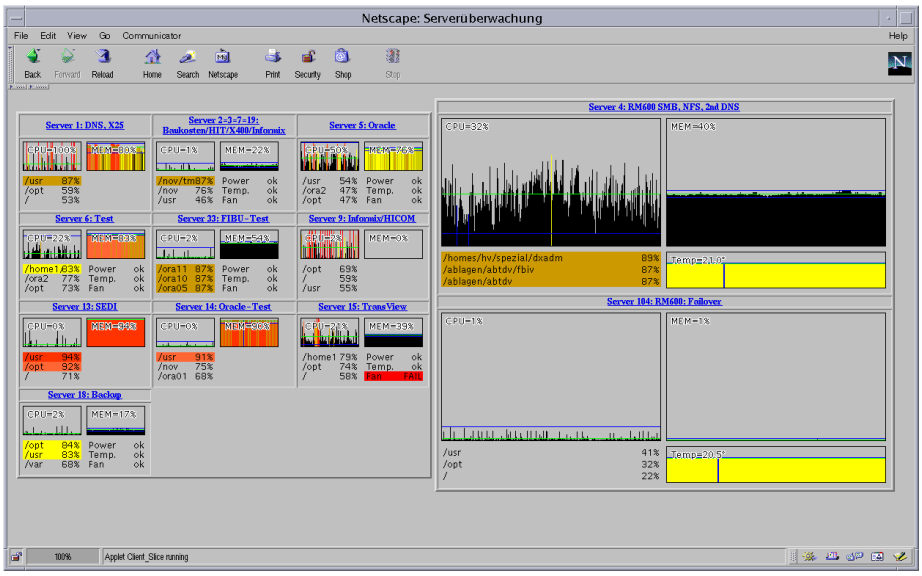


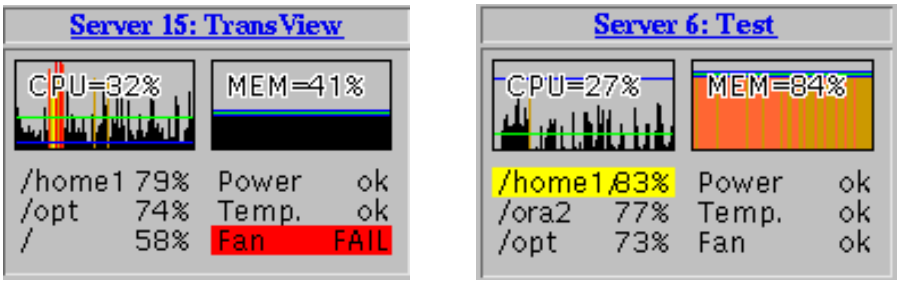
Abbildung 03-2: Schnappschuss aus einem Praxiseinsatz bei der LVA Rheinprovinz

Der Prototyp umfasst die Überwachung und Visualisierung der Parameter

- Erreichbarkeit im Netz (generisch für alle Plattformen per ICMP)
- CPU-Auslastung in %
- Auslastung des Speichers
- Umgebungstemperatur (realisiert auf RM600 E70 unter Reliant Unix)

@ Hardwarestörungen (Stromversorgung, Temperatur, Lüfter; realisiert auf RM400 C90 unter Reliant Unix)

@ Filesystemauslastung der „vollsten“ Partitionen



Hervorgehobene Fehleranzeige Tendenzerkennung durch Farben

Abbildung 03-3**:** Lüfterausfall; Verzeichnisfüllgrad überschreitet eingestellten Schwellwert

Aufgrund der in der Entwicklungsumgebung des Prototypen verfügbaren Serversysteme wurden alle Managementvorgänge zunächst unter Reliant Unix entwickelt. Mit wenigen Ausnahmen lassen sich ohne Veränderungen auch Solaris-, Linux- und sonstige Unix-Server überwachen. Abweichende Mechanismen zur Bestimmung der Parameter können leicht in die vorhandenen Serverprozesse integriert werden.

Enterprise-Server von SUN Microsystems gestatten z.B. die separate Visualisierung der Umgebungstemperatur, der Temperatur jeder vorhandenen CPU, jedes Systemboards, der jeweiligen Versorgungsspannungen und so weiter. Selbst der Zustand von Speichermodulen, die derzeitige Geschwindigkeit der CPU- und Netzteillüfter sowie die LED-Anzeigen der Gerätefronts sind abfrag- und damit visualisierbar. Ein Applet kann beispielsweise eine SUN Enterprise 450 grafisch darstellen und durch Wiedergabe der LED-Anzeigen im Webbrowser echtes “Lights Out Management” realisieren. Dieses ermöglicht eine Systemüberwachung aus „beliebiger“ Entfernung ohne Kontakt zum Server selbst.

03.05 Installation und Betrieb

Auf der Serverseite reduziert sich die Installation des Managementsystems auf das Einspielen der Software und das Starten der Managementserver. Auf einem Multiprozessorsystem findet die Aufteilung der Managementprozesse auf die Prozessoren automatisch durch das Betriebssystem statt. Eine Aufteilung auf mehrere Managementserver ist manuell dadurch

möglich, dass Teile der Prozesse jeweils auf unterschiedlichen Maschinen gestartet werden. Eine Koexistenz desselben Managementprozesses auf verschiedenen Servern ist ohne Einschränkung gewährleistet. Wird beispielsweise die CPU-Überwachung intensiver als die Speicherüberwachung genutzt, kann ein Managementserver CPU- und Speicherüberwachung zur Verfügung stellen, während ein zweiter Server ausschließlich die CPU-Überwachung bedient. Durch die beliebige Verteilung und Koexistenz von Diensten und Managementservern wird eine große Flexibilität in den Bereichen Inbetriebnahme, Betrieb und Erweiterung gewährleistet.

Zur Inbetriebnahme der Clients ist das Anpassen der jeweiligen HTML-Seite erforderlich. Je nach Wunsch können mehrere Clients über individuelle oder auch eine gemeinsame HTML-Seite auf das System zugreifen. Innerhalb dieser HTML-Seite sind die Adressen des/der Managementserver(s) sowie die zu überwachenden Systeme und Parameter spezifiziert. Das Aussehen der Seiten sowie eine eventuelle Aufteilung auf mehrere Seiten ist nicht fest vorgegeben, sondern kann beliebig dem gewünschten Layout angepasst werden. Gängige HTML-Editoren und Autorensysteme können verwendet werden, sofern sie die Integration der die eigentliche Funktion bereitstellenden Java-Applets gestatten.

Die flexible Darstellung mit Hilfe von HTML-Seiten ermöglicht eine nahtlose Integration des Systems in vorhandene Überwachungsmechanismen. Das Aussehen der Seiten, in die die Überwachungsapplets integriert werden, ist beliebig. Auch eine Kombination mit anderen Systemen, beispielsweise mit vorhandenen Intranet-Seiten, ist problemlos möglich.

Weiterer Vorteil ist der flexible Einsatz von jedem Arbeitsplatz aus. Schränkt man den Zugriff auf die Managementseiten nicht ein, kann sich ein Administrator bei Bedarf von jedem Arbeitsplatz aus einen schnellen Überblick über den Systemzustand verschaffen. Geeignete Autorisierungs- und Authentisierungsmaßnahmen vorausgesetzt, ist auch eine Anbindung an das Internet risikolos möglich, die es jedem Berechtigten ermöglicht, nach Eingabe von Benutzernamen und Kennwort von jedem Ort der Welt aus auf das System zuzugreifen. Fachpersonal ist nicht mehr räumlich gebunden. Überwachungsinfrastruktur muss nicht mehr dezentral, wie z.B. in jeder Außenstelle, vorgehalten werden, sondern qualifiziertes Personal kann zentral konsolidiert werden.

03.06 Ausblick

Der Prototyp ist voll funktionsfähig und wird im praktischen Einsatz genutzt. Veränderungen sind jedoch sowohl konzeptioneller Art als auch in Form von Erweiterungen für andere zu überwachende Betriebsparameter oder Plattformen denkbar.

Als Erweiterung ist eine langfristige Protokollierung und Archivierung der gemessenen Werte wünschenswert, um nachträglich kritische Situationen zu analysieren und zu bewer-

ten. Zu diesem Zweck ist es hilfreich, jederzeit die dargestellten Werte kommentieren zu können, um z.B. besondere Betriebsbedingungen zu dokumentieren.

Hierzu bietet sich die Integration einer Datenbank an, die in konfigurierbaren Zeitabständen den derzeitigen Systemzustand persistent vermerkt und für spätere Krisen- und Trendanalysen bereithält.

Die Liste der überwachbaren Betriebsparameter ist vollständig kaum zu erfassen. Hierzu nur einige Beispiele:

- **CPU:** Auslastung, Temperatur, Versorgungsspannung, Cache-Hit-Rate
- **Speicher:** Auslastung, Temperatur, Versorgungsspannung, ECC-Fehler, defekte Module, vom System ausgemappte Module
- **Festplatten:** genutzte Kapazität, genutzter Durchsatz pro Gerät/IO-Kanal/Controller/..., Temperatur, Versorgungsspannung, Zustand von RAID-Systemen, von Hot-Swap- und Hot-Standby-Systemen
- **Netzwerk:** Zustand der Netzverbindung: Link Status, Verbindungstyp voll/halbduplex, 10/100/1000MBit, Trunk-Status, Netzdurchsatz, Netzstörungen, Kommunikationspartner sortiert nach Volumen
- **Ausnutzung logischer Betriebsmittel**: Filehandle, Shared Memory, Semaphoren, Prozesse pro Benutzer uvm.
- **Controlling:** Anzahl Benutzer

Alle Detailinformationen können zu einer "Ampelfarbe" konsolidiert werden. Wenn ein Subsystem insgesamt fehlerfrei ist, können Detailinformationen unterdrückt werden. Erst bei vom Sollzustand abweichenden Parametern sollten diese angezeigt werden, um eine Reizüberflutung zu verhindern.

03.07 Fazit

Im praktischen Einsatz hat sich gezeigt, dass ein professionelles Systemmanagement ohne Installation proprietärer Clientsoftware realisierbar ist. Die Auslegung der grafischen Oberfläche als HTML-Seite ermöglicht eine neue Dimension der Flexibilität. Die fortlaufend aktualisierte Visualisierung der kritischen Betriebsparameter kann in beliebige Webseiten integriert werden. Größe und Aktualisierungsintervall sind frei wählbar. Eine entsprechende Freischaltung per Webserver und Firewall vorausgesetzt, ist eine Überwachung über beliebige IP-basierte WANs möglich. Dies ist insbesondere für die Organisation von Bereitschaft und Outsourcing interessant.

In der Praxis ist der vorgestellte Managementleitstand „Light“ meist der erste Indikator bei der Früherkennung von Problemen. Bei eingehenden Problemmeldungen hilft das System bei der Zuordnung des Problems auf die betroffenen Server. Lastsituationen lassen sich schnell erfassen, Routinevorgänge bei der Systemüberwachung sind nun auf einen Blick möglich.

Das System ermöglicht eine effizientere Ausnutzung der knappen und daher teuren Ressource Personal. Die vorhandenen Kräfte können schneller und mit geringerer Fehlerquote den aktuellen Systemzustand erfassen, bewerten und die richtigen Schlüsse ziehen.

03.08 Danksagungen

Eine wertvollen, methodischen und konzeptionellen Unterbau für die Bereitstellung des beschriebenen Leitstandes leistete das BMBF-Förderprojekt MAPKIT (Förderkennzeichen: 01 IS 801 B).

Weiterhin danken wir allen Kollegen aus den beteiligten Organisationen, die bei der Bearbeitung dieses Projektes mitgewirkt haben, insbesondere möchten wir hier die Kollegen Dieter Potemba und Sascha Laatsch (LVA Rheinprovinz) und die Kollegin Katja Gutsche (Materna GmbH) nennen.

KAPITEL 04 MIT LAST- UND PERFORMANCE-TESTS ZUM EINSATZREIFEN SYSTEM

HADI STIEL

04.01 Die Aufgabe

Das Bundesamt für die Anerkennung ausländischer Flüchtlinge (BAFI) stand Ende der 90ziger Jahre vor einer großen Herausforderung: Das Asylverfahren der Bundesrepublik Deutschland in voller Breite zu modernisieren. Dazu sollte ein umfassendes Dokumentenmanagementsystem (DMS) inklusive integrierter Vorgangsbearbeitung und optischer Archivierung zum Einsatz kommen. Denn das bisherige Host-basierende ASYLON, ohne DMS-Perspektive, ist den hohen Anforderungen in der Nürnberger Zentrale und den 33 bundesweit verteilten Außenstellen nicht mehr gewachsen. Zudem sind das Bundesverwaltungsamt mit dem Ausländerzentralregister (AZR) in Köln, das Bundeskriminalamt (BKA) in Wiesbaden sowie die Bundesbeauftragten für Asylangelegenheiten (BbfA) in die elektronische Vorgangsbearbeitung einzubinden. Summa summarum 1.500 Teilnehmer sollen so künftig an den Vorzügen eines leistungsfähigen Dokumentenmanagements partizipieren.

Die Akten der Asylsuchenden sollen weiterhin in der Nürnberger Zentrale residieren, künftig abgelegt auf Reliant Unix-Servern innerhalb einer Oracle-Datenbank Version 8i. Von dort aus ist der Fluss der Dokumente per Workflow über Index-Dateien, ebenfalls hinterlegt in der Oracle-Datenbank, zu steuern. Diese Vorgangsbearbeitung wird über Kernfunktionen des Produktes DOMEA von SER Floware in Salzburg abgebildet, erweitert um Individual-Software von Siemens Business Services (SBS), Berlin. Microsoft Exchange/Outlook ist als netzweiter Terminkalender für die Planung der Anhörungstermine einzusetzen. Durch die zusätzliche Einbindung des Fax-Servers von Right Fax in Exchange sollen künftig auch den externen Stellen ohne E-Mail-Anschluss eine Einstiegsschnittstelle ins Dokumentenmanagement geboten werden. Die abgeschlossenen Akten sollen in einem optischen Archiv, realisiert über einen Dokumenten-Server von CE und angesiedelt auf einem Windows 2000-Rechner, ebenfalls in Nürnberg Platz finden. Dieses Archiv soll im Zugriff der berechtigten Mitarbeiter in der Zentrale sowie in allen 33 Außenstellen liegen.

Siemens Business Services war die komplette Abwicklung des Projektes übertragen worden, einschließlich Piloten und Roll-out.

04.02 Hochgesteckte Ziele

Frank-Manuel Häusler-Kordt, Projektleiter bei Siemens Business Services in Berlin und technischer Projektleiter des DMS-Projektes IT2000 beim Bundesamt für die Anerkennung ausländischer Flüchtlinge, fasst die hochgesteckten Ziele des Vorhabens zusammen:

- Die Effektivität der Sachbearbeitung steigern,
- dadurch die Bearbeitungszeiten verkürzen,
- die Flexibilität und Wartung des Gesamtsystems verbessern sowie
- Raum und Botengänge durch die Ablösung des umfangsreichen Papierarchivs durch ein optisches Archiv einsparen.

„In unsere Verantwortung fällt zudem die Einbindung der bestehenden Host-Verfahren des Bundesverwaltungsamts und des Bundeskriminalamts sowie die Migration der Daten aus dem Altsystem ASYLON“, so Häusler-Kordt. „Dazu müssen wir für einen zwischenzeitlichen Parallelbetrieb die ASYLON-Daten auf die Daten des neuen Asylverfahrens synchronisieren.“ Um den bestehenden Papierberg über die Zeit abzubauen, sind diese Akten sukzessiv zu scannen und so allmählich ins optische Archiv zu überführen. Neuanträge, einschließlich des Passbildes und der Fingerabdruckblätter, sind bei Eingang in den Außenstellen sofort per Scanner zu erfassen und zu identifizieren, damit sie nahtlos ins zentrale Archivsystem einfliessen können. Speziell für den Scan der Personenfotos wird jede Außenstelle mit einer digitalen Kamera ausgerüstet.

04.03 Design und Implementierung von Szenarien

„Um die Vorgangsbearbeitung in geregelte Bahnen zu lenken, mussten wir erst einmal alle Workflow-Prozesse entlang der geplanten Bearbeitungsabläufe per Prozessdesigner von SER Floware designen und implementieren“, verweist der Projektleiter auf die Komplexität des Workflows. Dazu waren für die Bearbeitung der Asylanträge Ablaufmodelle zu berücksichtigen, die sowohl die Außenstellen als auch die beiden anderen Bundesämter einschlossen. Mit allen Sonderfällen waren dazu über vierzig Szenarien notwendig. Bei der Entwicklung dieser Szenarien musste zudem darauf geachtet werden, dass Medienbrüche durch den Einsatz von einheitlichen Bearbeitungswerkzeugen soweit wie möglich vermieden wurden. In diesem Kontext spielt der Dokumenten-Server eine wichtige Rolle, auf den via Windows NT 4.0-Clients zugegriffen wird. Dafür wurde speziell von SBS eine Zugriffslösung entwickelt. Als allgemein verbindliches Dokumentenformat gilt netzweit WinWord 7.0.

04.04 Strukturierung und Verfügbarkeit

Die Server für die Oracle-Datenbank, die Vorgangsbearbeitung und das Archivsystem wurden in eine zentrale (Server-Dienste) und dezentrale Ebene (Client-Dienste) gegliedert und hoch verfügbar ausgelegt. Im Fall der Reliant Unix-Server mit der Oracle-Datenbank und dem Workflow-System sorgt Reliant Monitoring System (RMS) für eine ständige Überwachung der Hauptrechner und schaltet im Problemfall auf den Ersatzrechner. Parallel spiegelt RMS permanent die Platten der Hauptrechner und garantiert so, dass auch die Ersatz-Server einen stets aktuellen Datenbestand vorhalten. Im zweiten Fall, dem Dokumenten-Server, wird die Hochverfügbarkeit über Windows 2000-Clustering erreicht. Die DMS- und Archivdienste selbst wurden als objektorientiertes System konzipiert und in Schichten, Packages und Objektklassen strukturiert. Die Kommunikation zwischen den Schichten wird über Microsofts Component Object Model (COM) realisiert. „Daneben haben wir zentral wie dezentral jeweils einen Kommunikations-Server vorgesehen“, beschreibt Häusler-Kordt. „Darüber sollen künftig Dienste wie Drucken im Hintergrund, elektronischer Versand sowie das automatische Routen der Akten mit den WinWord 7.0-Dokumenten und den Referenzen auf die archivierten Dokumente abgewickelt werden.“

04.05 Durch realitätsnahe Simulation zur Praxisreife

Um die Praxisreife der komplexen Vorgangsbearbeitung innerhalb aller Szenarien frühzeitig nachzuweisen, hatte sich das Bundesamt für die Anerkennung ausländischer Flüchtlinge auf Anraten von Siemens Business Services etwas besonderes ausgedacht: eine realitätsnahe Simulation einiger Workflow-Prozesse. Sie waren dazu Anfang Januar 2000 binnen acht Tagen im HS LAN Center von Siemens Business Services in Paderborn nachgebildet worden.

Abbildung 04-1: Blick ins LAN-Center von SBS in Paderborn

Aufgabe dieser Generalprobe: die künftigen Lasten auf den Servern zu simulieren, um so mögliche Engpässe innerhalb des Asylverfahrens aufzuzeigen. „Dazu haben wir im Vorfeld der Simulation die zu erwartenden Lasten gemeinsam mit dem Bundesamt für die Anerkennung ausländischer Flüchtlinge für unterschiedliche Zeitschienen bestimmt, differenziert in Recherchen, Anlegen und Bearbeiten von Geschäftsvorfällen sowie das Anlegen und Bearbeiten von Dokumenten“, skizziert Michael Lücke, Senior Performance Consultant bei Siemens Business Services in Paderborn und zuständig für die umfassende Simulation. „Die unterschiedlichen Lastmix-Aufkommen haben wir im realen Kontext von 1.500 Benutzern simuliert. Dies zu erreichen, mussten wir den 240 PCs im HS LAN Center die sechs- bis siebenfache Last eines typischen Benutzers auferlegen.“

Entwicklung und Aufbau der Simulationsumgebung

Für die umfassende Simulation der künftigen Vorgangsbearbeitungsprozesse beim Bundesamt für die Anerkennung ausländischer Flüchtlinge war eine gründliche technische Vorarbeit notwendig. Detlef Grothe, Leiter des Testlabors und neben Lücke zuständig für die Simulation, um darüber im einzelnen Lastverhalten und Antwortzeiten im künftigen Vorgangsbearbeitungssystem nachzuvollziehen, beschreibt den Testaufbau:

„Die Lastgeneratoren wurden mittels Visual Test von Rational erstellt, um darüber die Benutzereingaben an der Bedieneroberfläche zu erzeugen. Um die für das BAFI typischen PCs, insgesamt 240, zu installieren, haben wir eigene Verfahren eingesetzt. Sie ermöglichen uns, in kurzer Zeit von zentraler Stelle aus eine bedienerlose Installation von Betriebssystem, Anwendungen, Diensten, Tools sowie der Lastgeneratoren durchzuführen. Auf gleichem Wege haben wir die angepasste Steuerungsumgebung implementiert. Sie erlaubt uns, die diversen Anwendungsfälle von zentraler Stelle zu starten und kontrolliert mit unterschiedlichen Lastaufkommen ablaufen zu lassen. Auf diese Weise konnten wir beispielsweise auch die 1.500 PCs über die realen 240 PCs nachbilden.

.Das alles zusammen hilft uns, die Zeit für den Testaufbau auf ein Minimum zu reduzieren und dennoch eine Simulation hart an der Einsatzpraxis zu fahren. Server-seitig haben wir dazu eine Messumgebung auf dem Workflow- und dem Datenbank-Server (Oracle 8.05 / Unix) eingerichtet. Darüber können unter verschiedenen Lastsituationen die wesentlichen Systemressourcen dieser Rechner wie CPU, Festplatten, Hauptspeicher, Netzzugang überwacht und diese Werte gleichzeitig protokolliert werden. Das Antwortzeit-Verhalten in den unterschiedlichen Lastsituationen wurde stellvertretend für alle simulierten 1.500 PCs an einem ausgezeichneten Client nachvollzogen. Alle 240 PCs wurden entsprechend der anvisierten Installation beim BAFI an LAN-Segmente mit 10 und 100 Mbit/s angeschlossen, die via Switch-System mit dem Datenbank-Server am Backbone (Gigabit Ethernet) verbunden waren."

Während der Generalprobe wurden dann über eine speziell auf den Servern eingerichtete Messumgebung für die Performance wichtige Systemressourcen wie CPU, Festplatte, Hauptspeicher und Netzzugang überwacht und alle Messdaten protokolliert. Parallel wurde das Antwortzeit-Verhalten an einem ausgezeichneten Client ermittelt, der dazu repräsentativ für alle anderen Arbeitsstationen mit allen für den Einsatz notwendigen Anwendungen und Schnittstellen ausgestattet wurde. Lücke: „Auf diese Weise konnten parallel zu den generierten Lastvolumina die Antwortzeiten speziell kritischer Transaktionen wie *Business Case anlegen* und *versenden* exakt vermessen werden. Das wiederum ermöglichte uns, die Dienstgüte kompletter Vorgangsbearbeitungsabläufe exakt vorherzusagen."

Alle Mess- und Analyseergebnisse wurden in einem umfangreichen Messbericht zusammen gefasst. Er enthält auch eine Liste aller im Rahmen der Simulation verwendeten Kon-

figurationsdateien sowie Tuning-Vorschläge für einen optimalen Betrieb der Workflow-Prozesse. Herr Lücke fasst die wichtigsten Schwächen und ihre Behebung zusammen:

- Beim Anlegen großer Datenbestände (ab eine Millionen Akten- und Dokumentenverweise) ging der Durchsatz der Oracle-8-Datenbank merklich in die Knie: Diese Engpasssituation wurde auf Basis der Messprotokolle gezielt durch eine Index-Optimierung innerhalb der Oracle-Datenbank aufgelöst.
- Mit Blick in die weitere Zukunft und einer angenommenen Durchsatzrate von 4.500 Vorgängen pro Stunde erwiesen sich die vorgesehenen Systemressourcen in den Servern als zu gering: Durch eine angemessene Aufrüstung bieten jetzt die Server genügend Potenzial auf Jahre hinaus.
- Bei hohen Lasten wurden Schwächen in der vermessenen Version der DOMEA-Anwendung identifiziert, zurückzuführen auf Designprobleme innerhalb der Software. Das hatte in diesen Lastsituationen inakzeptable Antwortzeiten zur Folge: Diese Schwäche können wir mit der Folgeversion dieser Anwendungen ausräumen, die bereits Anfang 2001 auf den Markt kommen wird.

04.06 Pilotphase und Roll-out

Gut vorbereitet, kann es jetzt zügig an die weiteren Schritte des IT2000-Projektes gehen. Die Pilotinstallation zwischen der Zentrale in Nürnberg und der Außenstelle in Dortmund ist für Juli 2001 geplant. In diesen Piloten, um damit die Gesamtinstallation von Grund auf auf Herz und Nieren zu prüfen, werden auch das Bundesverwaltungsamt und Bundeskriminalamt einbezogen werden, inklusive Synchronisation der Daten aus dem Alt-Verfahren ASYLON auf den neuen Datenbestand. Im vierten Quartal 2001 wird dann der komplette Roll-out über alle 34 Standorte über die Bühne gehen. „Danach wird die Sachbearbeitung rund um die Anerkennung von Asylsuchenden eine neue Dienstleistungsdimension erreichen“, ist sich das BAFl sicher.

KAPITEL 05 LAST- UND PERFORMANCE-TESTS AM BEISPIEL IT2000

REINHARD BORDEWISCH, MICHAEL LÜCKE

05.01 Einführung und Motivation

In modernen IT-Strukturen bestehen starke Interdependenzen zwischen den Aspekten der Verfügbarkeit, der Zuverlässigkeit und der Performance. So kann es insbesondere in Hochlastfällen zum Reorganisieren von Netzwerkkomponenten und zum Rebooten von Server-Systemen sowie sogar zum Systemstillstand kommen. Deshalb werden insbesondere in vernetzten heterogenen Strukturen vor der Inbetriebnahme des Realbetriebs Last- und Stresstests durchgeführt, um das Systemverhalten unter Spitzenlast zu analysieren und zu bewerten.

Diese Tests erstrecken sich i.d.R. auf die Einbeziehung der Netzwerke und deren Komponenten sowie einer großen Anzahl Clients. Mit der Durchführung von Lasttests wird für den Anwender ein Mehrwert generiert, der sich wie folgt darstellt:

- Bedarfsgerechte Dimensionierung der Server-Systeme (Rightsizing) vor dem Realbetrieb (auch für unterschiedliche Konfigurationen),
- Aussagen im Wesentlichen zum CPU-, Platten- und Hauptspeicherbedarf jeweils für verschiedene Lastfälle (applikations-spezifisch),
- Möglichkeit der Ermittlung des Ressourcenbedarfs bei unterschiedlichen Lastzusammenstellungen,
- Analyse und Eliminierung von Engpässen vor der Aufnahme des Produktivbetriebs (keine Störungen im Realbetrieb),
- Ermittlung des Systemverhaltens (Verfügbarkeit, Zuverlässigkeit, Performance) insbesondere in Hochlastsituationen vor Aufnahme des Realbetriebs („Stresstest“).

Daraus resultiert nicht nur eine Kostenverlagerung sondern auch eine Kostenreduzierung, da so die bedarfsgerechte Dimensionierung und die Aufnahme des störungsfreien Produktivbetriebs ermöglicht werden (vgl. Kasten).

Projekt: Last- und Performancetest IT2000

Kunde: „Bundesamt“

Ausgangssituation:

Kunde verlangt Leistungsnachweis bez. folgender Punkte:
- Können 4 500 Transaktionen pro Stunde verarbeitet werden?
- Kann Reliant UNIX die aufgerufenen Prozesse bearbeiten?
- Über welche CPU-Power muss der DB-Server verfügen?

Realisierung:

- Simulation von 1 500 Benutzern auf 240 physikalischen PCs
- Ermittlung von Durchsatzraten, CPU- und DB-Last mittels einer vollautomatisierten Test-, Mess- und Auswerteumgebung
- Überprüfung des Antwortzeitverhaltens

Kundennutzen:

- Funktionaler Test des Gesamtsystems
- Aufdecken von Designproblemen
- Nachweis über geforderten Durchsatz und Antwortzeiten
- Dimensionierung des Gesamtsystems
- Deutliche Performance-Verbesserungen durch DB-Optimierung

05.02 Automatische Lastgenerierung

Zur Analyse und Bewertung des Gesamtsystemverhaltens ist die Einbeziehung einer definierten Last unabdingbar. Die Umsetzung der realen Last erfolgt in Form von so genannten Lastprofilen. Dabei stehen in diesem Umfeld zwei Ansätze zur Verfügung, die auch kombinierbar sind. Zum einen kann die reale Last in synthetische Lastprofile umgesetzt werden, d.h. die Anwenderlast wird mittels vorgegebener Funktionen und Aufträge realitätsgetreu nachgebildet und unabhängig von der realen Applikation ausgeführt. Im anderen Fall erfolgt die Umsetzung der Anwenderlast in reale Lastprofile, d.h. es werden die Eingaben und das Verhalten der Benutzer im realen Applikationsumfeld synthetisiert, worauf sich im Folgenden die automatische Lastgenerierung bezieht.

(a) Anforderungen

Damit Messungen auf Basis von realen Lastprofilen in vergleichbaren Gesamtsystemen auch vergleichbare Ergebnisse liefern, sind bestimmte grundlegende Anforderungen zu erfüllen:

@ Generierbarkeit: Die vorgegebenen Anwendungsfälle müssen die Umsetzung in erzeugbare Lastprofile unterstützen, d.h. die Last kann nicht aus der Ausführung zusammenhangloser Transaktionen bestehen, sondern muss aus einer zusammenhängenden Abfolge von Benutzereingaben bestehen. Dieses Kriterium ist damit auch Voraussetzung, um ein reales Benutzerverhalten nachzuvollziehen.

@ Reproduzierbarkeit: Die Lastprofile müssen wiederholt ausführbar sein, um die Last für bestimmte Zeitspannen erzeugen zu können. Bei der Wiederholung ist zu vermeiden, dass aufgrund von Lokalitäten referierter Daten Messverfälschungen eintreten.

@ Überprüfbarkeit: Während der Erzeugung der Last ist die Aufnahme gewisser Messdaten (z.B. Durchsätze, Antwortzeiten) an den Erzeugungssystemen notwendig. Diese dienen einerseits zur Kontrolle der Einhaltung der vorgegebenen Lastdefinition und anderseits zur Quantifizierung der Auswirkungen der erzeugten Last auf das Verhalten des Gesamtsystems.

Zur statistischen Absicherung der verschiedenen Messfälle sollte die Wiederholung der Lastzyklen beliebig oft erfolgen können. So ist gewährleistet, dass bei verschiedenen Messungen mit identischem Lastprofil in gleicher Messumgebung vergleichbare Ergebnisse erzeugt werden ("Reproduzierbarkeit" der Ergebnisse). Damit ist die Notwendigkeit einer automatischen Lasterzeugung gegeben.

Die Durchführung von Messungen mit realem Personal in einem Umfeld von vielen Clientsystemen scheitert an deren Verfügbarkeit und den damit verbundenen Kosten. Aber auch bei einer kleinen Anzahl von Clientsystemen ist die händische Lasterzeugung mit Unwägbarkeiten verknüpft, da durch

@ fehlerhafte Eingaben ,

@ unkontrolliertes Zeitverhalten bei den Eingaben und

@ unterschiedliche Arbeitsweisen verschiedener Benutzer

keine kontrolliert ablaufende Lasterzeugung stattfindet. Somit ist nicht auszuschließen, dass Messverfälschungen durch unterschiedliche Belastungen im Gesamtsystem auftreten. Die Reproduzierbarkeit der Ergebnisse wäre nicht gewährleistet. Hinzu kommt, dass die Aufnahme von Antwortzeiten und Durchlaufzeiten parallel zu den Eingaben der realen Benutzer relativ schwierig ist.

Zur Verbesserung der Durchführung der automatischen Lasterzeugung wird die Anforderung nach Variabilität gestellt. Die Lastprofile sind so zu erstellen, dass durch Parametrisierung eine einfache und schnelle Ausführung verschiedener Lastprofile mit unterschiedlichen Durchsätzen für beliebige Zeitdauern erfolgen kann.

Dies bedeutet im Einzelnen, dass

1. verschiedene Transaktionsabfolgen erzeugt werden können.
2. die Erzeugungsgeschwindigkeit und das Durchsatzverhalten variiert werden kann, so dass hierüber die Simulation von mehreren Benutzern auf einem physikalischen Client ermöglicht wird.
3. die Dauer der Lasterzeugung gesteuert werden kann. Für die Durchführung von Messungen im eingeschwungenen Zustand ist es notwendig, jeweils eine Vor-, Mess- und Nachlaufphase bei der Lasterzeugung einzuführen.

Der Vorteil besteht darin, dass von außen gesteuert das Lastverhalten von der einen zur anderen Messung ohne Probleme schnell und einfach verändert werden kann (Kalibrierung von Anwenderlastprofilen). Entsprechend kann die Last auch kontinuierlich erhöht werden, um so komplette Messreihen für Dimensionierungsaussagen durchzuführen.

Für die Lastgenerierung auf sehr vielen lasterzeugenden Clients (> 100) ist eine zentrale Steuerungsinstanz erforderlich, welche die Messungen vorbereitet, durchführt und überwacht sowie nachbereitet.

Neben der Performance-Messung wird die automatische Lasterzeugung auch zur Testautomatisierung in der Quality Assurance eingesetzt. In diesem Zusammenhang sind die folgenden Punkte besonders wichtig:

- Fehler-Handling: Während der Testdurchführung ist es erforderlich, Abweichungen zum vorgegebenen Verhalten zu erkennen. Abhängig von der vorliegenden Situation muss eine entsprechende Reaktion erfolgen (eventuell sogar der Testabbruch).
- Vollständigkeit: Durch die Betrachtung des Gesamtsystems und aufgrund der Synthetisierung der Benutzer direkt an der Mensch/Maschine-Schnittstelle werden alle zu überprüfenden Systemkomponenten im Test berücksichtigt.
- Testabdeckung: Um einen hohen Abdeckungsgrad zu erreichen, müssen mindestens die wichtigsten Testfälle bei der Auswahl der Lastprofile berücksichtigt werden.

Jede Performance-Messung eines Gesamtsystems beinhaltet gleichzeitig einen funktionalen Test desselben. Deshalb ist die Berücksichtigung des Fehler-Handlings auch bei der Realisierung der Lastprofile zu Messzwecken eine unabdingbare Anforderung. Die zeitgenaue

Protokollierung des Fehlverhaltens ist für die Interpretation der Ergebnisdaten unbedingt erforderlich.

Mit Hilfe der automatischen Lasterzeugung wird erreicht, dass definierte und reproduzierbare Lastprofile ohne großen personellen Aufwand zur Bestimmung und Analyse des Verhaltens des zu betrachtenden Gesamtsystems erzeugt werden können.

05.03 Durchführung von automatisierten Last- und Performance-Tests

(a) Aufgabenstellung

In vernetzten IT-Umgebungen werden Last- und Performance-Tests zweckdienlich mit einer sehr großen Anzahl (>> 100) von Clients durchgeführt, um so möglichst realitätsnah und vielseitig Lasten erzeugen zu können. Insbesondere ist die Einbeziehung des Netzwerkes bei solchen Tests ein nicht zu vernachlässigender Aspekt, dem nur mit der Beteiligung von sehr vielen Netzknoten (Clients) Rechenschaft geleistet wird. Erst die Möglichkeit, an vielen Clients Lasten zu erzeugen, die alle nahezu gleichzeitig oder auch nur sporadisch auf das Netzwerk und die Server-Systeme zugreifen, können Funktionalität und Performance des vorhandenen Netzes und somit das Verhalten des Gesamtsystems entsprechend analysiert und bewertet werden. Die Durchführung von Last- und Performance-Tests mit vielen Client-PCs macht es erforderlich, Mechanismen zur definierten Ansteuerung und Überwachung der Clients zur Verfügung zu stellen.

Hierbei sind zu nennen:

- die Verteilung der Skripte, Programme und Eingabe- oder Konfigurationsdateien
- die Zuteilung bestimmter Rollen in dem Last- oder Performance-Test (z.B. "Last-PC", der lediglich der reinen Lasterzeugung ohne Messung dient)
- die Auswahl und Parametrisierung verschiedener Lastfälle
- die gezielte (synchrone) Startaufforderung zur Lasterzeugung an die PCs
- das Einsammeln von erzeugten Ausgabedateien und erhobener Messergebnisse von den PCs
- die Überprüfung der korrekten Lastdurchführung
- die Auswertung der Messdaten

Des Weiteren müssen diese Mechanismen automatisch ablaufen, denn es ist sehr aufwändig, händisch alle Clients derart vorzubereiten, dass sie an dem Test partizipieren können. Andererseits lassen sich ohne diesen Automatismus gewisse Tests gar nicht oder nur mit

sehr hohem Personalaufwand realisieren. Dazu ist ein Steuerungsprogramm erforderlich, das die Durchführung der Last- und Performance-Tests von einer zentralen Instanz steuert.

(b) Funktionalitäten des Steuerungsprogramms

- Es steht eine Synchronisierungsmöglichkeit zur Verfügung.
- Kommandos werden an den Client- und an den Serversystemen von der zentralen Steuerungsinstanz initiiert ausgeführt.
- Dateien sind zwischen der zentralen Steuerungsinstanz und den Clients sowie Servern in beide Richtungen übertragbar.

(c) Zentrale Steuerungsinstanz und Ablaufumgebung

Die Realisierung basiert auf einem im Rahmen von MAPKIT entwickelten Werkzeug, mit dem Zeichenketten zentral von einem System über Netzwerk zu anderen Systemen übertragen werden können. Dieses Werkzeug beinhaltet einen Sender- und einen Empfängerteil.

(d) Schematischer Ablauf eines Last- und Performance-Tests

Vorbereitung

- Erstellen einer Liste mit am Lasttest beteiligten Clients und einer Liste der beteiligten Server-Systeme
- Bereitstellen der für den Lasttest notwendigen Dateien und Programme - insbesondere der Lasterzeugungsumgebung
- Übertragung der Kommandofolge an Server und Clients zum Kopieren der jeweils notwendigen Dateien
- Übertragung der Kommandofolge an Server und Clients zur Durchführung server- und clientspezifischer Maßnahmen (z.B. Anmeldung, Rücksetzen der Testumgebung auf einen definierten Stand, ...)

Messung

- Übertragung der Kommandofolge an die Clients und ggfs. Server zum Start der Lasterzeugung (Verzögerungszeiten möglich)
- Übertragung der Kommandofolge an Server und ggfs Clients zum Start der Messwerkzeuge
- Abwarten der Messdauer

Nachbereitung

- Überprüfung aller beteiligten Systeme auf Beendigung des Tests
- Übertragung der Kommandofolge an Server und Clients zum Einsammeln der Ergebnis- und Protokolldateien
- Vorauswertung der Ergebnisdateien an der Steuerungsinstanz

05.04 Last- und Performancetest ATLAS

Im Zuge der ATLAS-Serverkonsolidierung (Übergang von derzeit 25 gleichzeitig aktiven Benutzern auf mehr als 250 an einem System) wurden Fragen aufgeworfen, die sowohl funktionaler Art waren als auch Aussagen zum Systemverhalten beinhalteten:

- Kann ATLAS mehr als 250 Clients handhaben?
- Wie sind die Antwortzeiten?
- Wie sind die Server-Systeme und das Netzwerk (LAN- und WAN-Strecken) zu dimensionieren?
- Wo sind Engpässe zu erwarten? Wie können diese behoben werden?

Zur Absicherung der Aufnahme eines störungsfreien Produktivbetriebs war es deshalb erforderlich, einen entsprechenden Last- und Performance-Test durchzuführen.

Dabei wurden im SBS HS LAN Center in Paderborn 263 Client-PCs und unterschiedliche Server-Systeme (DB-Server (Solaris), Kommunikations-, Dienstleistungs- und PDB-Server (Reliant UNIX)) mit der realen ATLAS-Applikation installiert und die LAN-/WAN-Infrastruktur realitätsgetreu nachgebildet (vgl. Abbildung 05-1). Die Steuerung der Applikation erfolgte mittels des Treibersystems SQA-Robot.

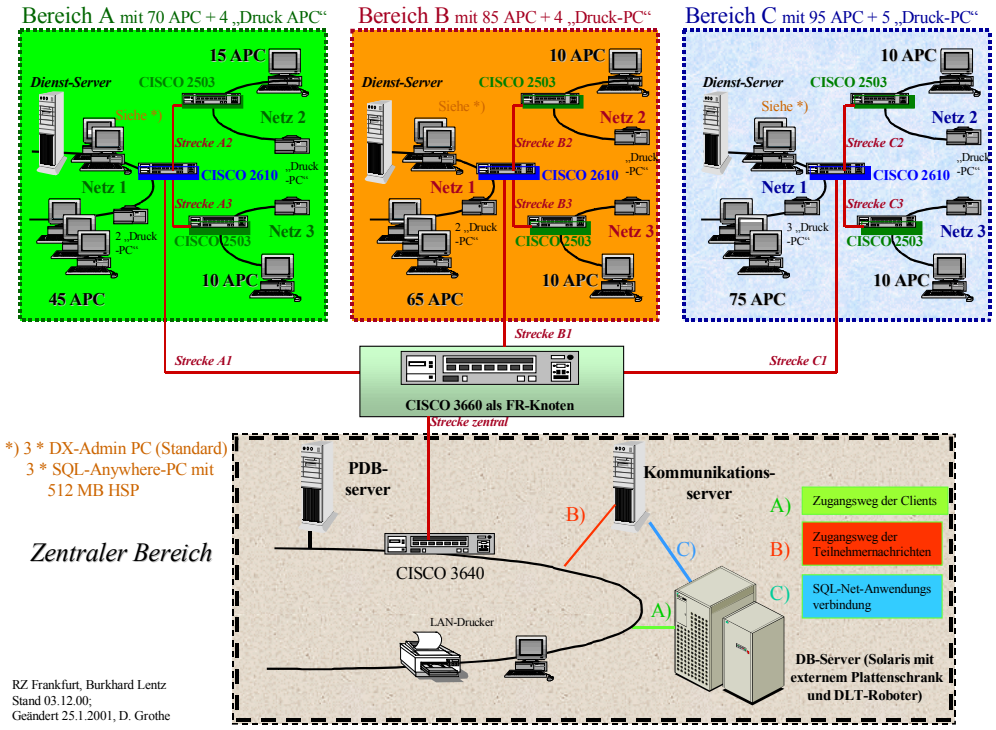


Abbildung 05-1: Aufbau der Messanordnung im Testlabor Paderborn

Es wurden unterschiedliche Messwerkzeuge eingesetzt:

@ Überwachung der Dienstgüte (Antwortzeiten) am Client mit dem *Referenz-PC* und SQA-Robot

- @ Ermittlung der Belastung der Server-Systemkomponenten (CPUs, Hauptspeicher, Disk-I/O-Subsystem, Netzzugang) mit PERMEV (Solaris) bzw. PERMEX (Reliant Unix)
- @ Ermittlung der Netzlasten mit verschiedenen Netzwerkanalysatoren

Fazit: Aufgrund der Komplexität der gesamten IT-Infrastruktur mussten zunächst umfangreiche qualitätssteigernde Maßnahmen durchgeführt werden. So verlagerte sich der Schwerpunkt der Aktivitäten immer mehr von den geplanten Performance-Messungen hin zu qualitätssichernden und stabilisierenden Maßnahmen. Damit wurden im Wesentlichen zwei globale Zielsetzungen erreicht: Einerseits konnte die Pilotierungsphase drastisch verkürzt werden und andererseits wurde so die störungsfreie Aufnahme des Produktivbetriebs ermöglicht (kein mangelbehaftetes Systemverhalten, kein Akzeptanzverlust).

Zusammenfassend sind folgende Ergebnisse und Kernaussagen zu nennen.

- @ Mit der ATLAS-Applikation können mehr als 250 parallel arbeitende Benutzer in dieser Infrastruktur zufriedenstellend bedient werden.
- @ Lediglich in den über WAN-Leitungen angeschlossenen Außenstellen (ohne die Dienstleistungs-Server) wurde das vorgegebene Antwortzeitverhalten in einigen speziellen Anwendungsfällen nicht erreicht.
- @ Die Server-Systeme sind bis auf den Kommunikations-Server gut dimensioniert. Die LAN- und WAN-Strecken verfügen (bis auf die o.g. Außenstandorte) über ausreichende Bandbreiten.
- @ Auf der Basis von Ergebnissen aus Detail- und Ursachenanalysen wurden Tuningansätze erarbeitet (u.a. für den Kommunikations-Server und die DB-Anbindung).

Teil VIII: Quellen

Niagara Fälle

Inhalt von Teil VIII

KAPITEL 01 REFERENZEN 3

01.01 LEHRBÜCHER (AUSWAHL) 3
01.02 FIRMEN 4
01.03 INSTITUTIONEN 5

KAPITEL 02 WEITERE RESSOURCEN IM WORLD WIDE WEB 6

KAPITEL 03 AUTOREN 7

KAPITEL 04 KONTAKTADRESSEN 10

KAPITEL 01 REFERENZEN

01.01 Lehrbücher (Auswahl)

@ W. Dirlewanger: Messung und Bewertung der DV-Leistung – Auf Basis der Norm DIN 66273, Hüthig-Verlag, Heidelberg, 1994

@ N. J. Gunther: The Practical Performance Analyst, McGraw-Hill Inc., New York, 1998.

@ G. N. Higginbottom: Performance Evaluation of Communication Networks, Artech House, 1998

@ R. Jain: The Art of Computer System Performance – Analysis, Techniques for Experimental Design, Measurement, Simulation and Modeling, Wiley, 1991

@ R. Klar et al.: Messung und Modellierung paralleler und verteilter Rechensysteme, B.G. Teubner; Stuttgart, 1995

@ H. Langendörfer: Leistungsanalyse von Rechensystemen – Messen, Modellieren, Simulation, Hanser Studienbücher, 1992

@ D. J. Lilja: Measuring Computer Performance – A Practitioner`s Guide, Cambridge University Press, 2000

@ E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik: Quantitative System Performance, Prentice-Hall Inc., New Jersey, 1984

@ M. Loukides: System Performance Tuning, O'Reilly-Verlag, 1995

@ D. A. Menascé, V. A. F. Almeida, L.W. Dowd: Capacity Planning and Performance Modeling – From Mainframes to Client-Server Systems, Prentice Hall, 1994

@ D. A. Menascé, V. A. F. Almeida: Capacity Planning for Web Performance – Metrics, Models, and Methods, Prentice Hall, 1998

@ D. A. Menasce, V. A. F. Almeida: Scaling for E-Business, Prentice Hall, 2000

01.02 Firmen

@ BMC: http://www.bmc.com/, Distributor von BMC Patrol und anderen ERP Produkten

@ Computer Associates Inc.: http://www.cai.com/, bekannt durch Unicenter TNG

@ Compuware: http://www.compuware.com, Distributor der EcoTools, welche auch den EcoPredictor zur Netzplanung umfassen

@ Hewlett-Packard: http://www.hp.com, Distributor der HP Tools, u.a. HP OpenView

@ HyPerformix: http://www.hyperformix.com, Distributor von SES/Strategizer, Schwerpunkt auf E-Commerce Performance Engineering

@ Institute for Computer Performance Management: http://www.iccmforum.com/, "a meeting place for people who care about performance"

@ Lund Performance Solutions: http://www.lund.com/, Dienstleistungen und Tools im Umfeld Capacity Planning

@ Make Systems: http://www.makesys.com/, "… provides the most advanced, most scalable network modeling and network design technology available today"

@ Materna Information and Communications: http://www.materna.de

@ Mercury Interactive: http://www-svca.mercuryinteractive.com/, Div. Tools u.a. im Bereich Web Capacity Management

@ Metron: http://www.metron.co.uk/ und www.metron.co.uk/technical/tech.html; Metron hat den Anspruch, der führende Software-Anbieter zu sein für „computer performance management and capacity planning"

@ Mil3: http://www.mil3.com, Distributor der OPNET-Werkzeuge

@ Siemens Business Services: http://www.siemens.com/sbs/de/offerings/services/st/pract_02d.html

@ Siemens SAP R/3: http://www.MyAMC.de, Homepage des Application Management Centers (AMC), welches u.a. den LNI (früher R/3 Live Monitor) umfasst

@ TeamQuest: http://www.teamquest.com/, Consulting und Tools

@ xware edv-beratung gmbh: http://xware.net/, R/3-Performance, insbesondere SAP Standard Applikations Benchmark

01.03 Institutionen

- @ Computer Measurement Group (CMG): http://www.cmg.org/
- @ Computer Measurement Central Europe (CMG-CE): http://www.cecmg.de, u.a. Arbeitsgruppe ”Planen, Modellieren, Messen”
- @ ACM Special Interest Group on Metrics of Computer Systems: http://www.sigmetrics.org/
- @ ACM Special Interest Group on Data Communication: http://www.acm.org/sigcomm/
- @ GI-Fachgruppe „Messung, Modellierung und Leistungsbewertung (MMB)“: http://www.informatik.unibw-muenchen.de/mmb/

KAPITEL 02 WEITERE RESSOURCEN IM WORLD WIDE WEB

- @ http://mapkit.informatik.uni-essen.de/, MAPKIT Projekt Server
- @ http://joe.lindsay.net/webbased.html, The Web Based Management Page
- @ http://www.capacityplanning.com/, CapacityPlanning.com was developed to provide users with information and white-papers about capacity management, measurement and planning as well as providing links to related sites
- @ http://www.bmc.com/products/whitepaper.html, Sammlung von Aufsätzen rund um die Produkte von BMC Software
- @ http://joe.lindsay.net/webbased.html, The Web Based Management Page
- @ http://www.informatik.uni-essen.de/Lehre/Material/CapPlan/, Materialien zur Lehrveranstaltung „Kapazitätsplanung und Leistungsbewertung“ am Institut für Informatik an der Universität Essen
- @ http://www.comnets.rwth-aachen.de/SFgate/cn-wais.html, Datenbank der RWTH Aachen zur Literaturrecherche im Bereich Modelle und Performance
- @ http://www.december.com/net/tools/index.html, umfangreiche Sammlung von Netzwerktools
- @ http://www.xware.net/sapbench/, SAP Standard Application Benchmark Database
- @ Tutorial on “Capacity Planning for Client/Server Systems”, Tutorial at Sigmetrics, 1995, Ottawa, Canada, (.ppt, 593K), http://www.cs.gmu.edu/faculty/menasce.html
- @ Capacity Planning for Web Performance, one day tutorial taught at George Mason University and at Usenix, 1998, http://www.cs.gmu.edu/faculty/menasce.html

KAPITEL 03 AUTOREN

@ BERNERT, TORSTEN
INSTITUT FÜR INFORMATIK, UNIVERSITÄT ESSEN

@ BERSCHNEIDER, JOSEF
SIEMENS BUSINESS SERVICES, MÜNCHEN

@ BORDEWISCH, REINHARD
SIEMENS BUSINESS SERVICES, PADERBORN

@ FLÜS, CORINNA
INSTITUT FÜR INFORMATIK, UNIVERSITÄT ESSEN

@ GLOBISCH, SIEGFIED
R+V-VERSICHERUNGEN, FRANKFURT/M

@ GRABAU, REINHARD
FUJITSU SIEMENS COMPUTERS, MÜNCHEN

@ GRADEK, STEFAN
FUJITSU SIEMENS COMPUTERS, PADERBORN

@ HEHN, MATHIAS
MATERNA INFORMATION & COMMUNICATIONS, DORTMUND

@ HINTELMANN, JÖRG
INSTITUT FÜR INFORMATIK, UNIVERSITÄT ESSEN
(SEIT OKT. 2000 ND SATCOM, FRIEDRICHSHAFEN)

@ HIRSCH, KLAUS
SIEMENS BUSINESS SERVICES, MÜNCHEN

@ KOWARSCHICK, CHRISTIAN
X-WARE EDV-BERATUNG , ÖSTRINGEN

@ LÜCKE, MICHAEL
SIEMENS BUSINESS SERVICES, PADERBORN

@ LÜHRS, JÜRGEN
MATERNA INFORMATION & COMMUNICATIONS, DORTMUND

@ MANZ-BOTHE, ACHIM
LVA RHEINPROVINZ, DÜSSELDORF

@ MÜLLER-CLOSTERMANN, BRUNO
INSTITUT FÜR INFORMATIK, UNIVERSITÄT ESSEN

@ PAUL, MICHAEL
FUJITSU SIEMENS COMPUTERS, BREMEN

@ PFISTER, JÜRGEN
FUJITSU SIEMENS COMPUTERS, MÜNCHEN

@ RISTHAUS, HEINER
MATERNA INFORMATION & COMMUNICATIONS, DORTMUND

@ SCHATEN, RONALD
MATERNA INFORMATION & COMMUNICATIONS, DORTMUND

@ SCHERNER, NORBERT
SIEMENS BUSINESS SERVICES, MÜNCHEN

@ SCHMUTZ, STEFAN
SIEMENS BUSINESS SERVICES, MÜNCHEN

@ SCHWÄRMER, BÄRBEL
SIEMENS BUSINESS SERVICES, PADERBORN

@ STEWING, FRANZ-JOSEF
MATERNA INFORMATION & COMMUNICATIONS, DORTMUND

@ STIEL, HADI
JOURNALIST, FRANKFURT

@ SYDOW, KATHRIN
SIEMENS ITS SIM ESM, FÜRTH

@ VILENTS, MICHAEL
INSTITUT FÜR INFORMATIK, UNIVERSITÄT ESSEN

@ WARLIES, KURT JÜRGEN
SIEMENS BUSINESS SERVICES, PADERBORN

@ WILHELM, KAY
INSTITUT FÜR INFORMATIK, UNIVERSITÄT ESSEN

KAPITEL 04 KONTAKTADRESSEN

Siemens Business Services GmbH & Co. OHG
Heinz-Nixdorf-Ring 1
D-33106 Paderborn
Tel.: + (05251) 815 460
Fax: + (05251) 815 503
Email: Reinhard.Bordewisch@pdb.sbs.de

Fujitsu Siemens Computers GmbH
FSC PO EP OS MF PM
Otto-Hahn-Ring 6
D-81739 München
Tel.: + (089) 636 47617
Fax: + (089) 636 41328
Email: Reinhard.Grabau@fujitsu-siemens.com

Siemens Business Services GmbH & Co. OHG
Otto-Hahn-Ring 6
D-81739 München
Tel.: + (089) 636 42145
Fax: + (089) 636 52888
Email: Klaus.Hirsch@mch20.sbs.de

Fujitsu Siemens Computers GmbH
FSC STM SAP TCS
Otto-Hahn-Ring 6
D-81739 München
Tel.: + (089) 636 45751
Fax: + (089) 636 44275
Email: Juergen.Pfister@fujitsu-siemens.com

Materna Information & Communications
Business Unit Information
Vosskuhle 37
D-44141 Dortmund
Tel.: + (0231) 5599 433
Fax: + (0231) 5599 272
Email: Franz-Josef.Stewing@materna.de

Universität Essen
Institut für Informatik
Prof. Dr. B. Müller-Clostermann
Arbeitsgruppe „Systemmodellierung“
Schützenbahn 70
D-45127 Essen
Tel.: + (0201) 183-3915, -3920
Fax: + (0201) 183-2419
Email: bmc@informatik.uni-essen.de